REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
18-12-2008	Final Report	15-Mar-2005 - 14-Sep-2008

4. TITLE AND SUBTITLE
Final Report: ATR Using Multiview Morphing

5a. CONTRACT NUMBER
W911NF-05-1-0090

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
611102
622303

6. AUTHORS
Mubarak Shah

5d. PROJECT NUMBER
622303

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
University of Central Florida
Office of Research
University of Central Florida
Orlando, FL 32826 -0150

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
46607-CS.1

12. DISTRIBUTION AVAILIBILITY STATEMENT
Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT
In 2005, reviewed current research literature in view morphing and matching for single views, and discussed a feature based approach to sparse view morphing that also shows promise for compression of the model data corpus. Worked on a new Bayesian approach for the problem from a mid-level segmentation problem. We obtained good results for feature-based morphing, as well as preliminary demonstration of the novel object representation.

In 2006, we completed work in automatic target recognition using video. Introduced a novel approach for adaptive video

15. SUBJECT TERMS
ATR, view morphing, object recognition

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Mubarak Shah
U	U	U	SAR		19b. TELEPHONE NUMBER 407-823-5077

Report Title

Final Report: ATR Using Multiview Morphing

ABSTRACT

In 2005, reviewed current research literature in view morphing and matching for single views, and discussed a feature based approach to sparse view morphing that also shows promise for compression of the model data corpus. Worked on a new Bayesian approach for the problem from a mid-level segmentation problem. We obtained good results for feature-based morphing, as well as preliminary demonstration of the novel object representation.

In 2006, we completed work in automatic target recognition using video. Introduced a novel approach for adaptive video registration. Formulated the image-registration problem as a region-partitioning problem. Developed an approach for video-based object recognition which doesn't require any knowledge of camera position or physical location of images with respect to each other. We exploited motion continuity in video and extend our algorithm to perform matching based on video input.

In 2007, we developed a novel method for object class detection based on 3D object modeling. We worked on object recognition based on correlation using morphing technique. We used two new methods for synthesizing new views of a known object so that the occluded features of the object can be inferred and incorporated into the recognition process.

List of papers submitted or published that acknowledge ARO support during this reporting period. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Jiangjian Xiao and Mubarak Shah, "Tri-view morphing", Computer Vision Image Understanding, Volume 96, Issue 3 , Pages 345-366, December 2004.

Number of Papers published in peer-reviewed journals: 1.00

(b) Papers published in non-peer-reviewed journals or in conference proceedings (N/A for none)

Number of Papers published in non peer-reviewed journals: 0.00

(c) Presentations

Number of Presentations: 0.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts): 0

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Jun Xie, Min Hu, and Mubarak Shah, "An Object Recognition Framework Using Unfolding Warping", International Conference on Pattern Recognition, December 2008.

Abhijit Mahalanobis, Philip Berkowitz, Mubarak Shah, Richard Sims, "View Morphing using Linear Prediction of Sub-Space Features", 41st Asilomar Conference on Signals, Systems and Computers, November 4-7, 2007.

Pingkun Yan, Saad M. Khan, and Mubarak Shah, "3D Model based Object Class Recognition in Arbitrary Views, IEEE International Conference on Computer Vision, October, 2007, Rio de Janeiro, Brazil.

Humera Noor, Shahid H. Mirza,Yaser Sheikh, Amit Jain, Mubarak Shah, "Model Generation for Video based Object Recognition", ACM MM 2006, Santa Barbara, CA, USA.

J. Xiao and M. Shah, "Automatic target recognition using multiview morphing", SPIE Automatic Target Recognition XIV Conference, 12-16 April 2004.

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts): 5

(d) Manuscripts

Number of Manuscripts: 0.00

Number of Inventions:

Graduate Students

NAME	PERCENT_SUPPORTED
Saad Khan	0.50
Fahd Rafi	0.50
Mikel Rodriguez	0.50
Imran Saleemi	0.50
Nazim Ashraf	0.50
Paul Scovanner	0.50
Ryan Faircloth	0.50
Vlad Reilly	0.50
Omer Orhan	0.50
Pavel Babenko	0.50
Phillip Berkowitz	0.25
Yaser Sheikh	0.50
FTE Equivalent:	**5.75**
Total Number:	**12**

Names of Post Doctorates

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Names of Faculty Supported

NAME	PERCENT_SUPPORTED	National Academy Member
Mubarak Shah	0.25	No
Kanad Biswas	0.25	No
FTE Equivalent:	**0.50**	
Total Number:	**2**	

Names of Under Graduate students supported

NAME	PERCENT_SUPPORTED
Emine McDonald	0.50
Ada Brewton	0.25
Andrew Miller	1.00
Sanda Lo	0.25
Qin Feng Chen	0.25
FTE Equivalent:	**2.25**
Total Number:	**5**

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period:	0.00
The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:.....	0.00
The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:.....	0.00
Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):	0.00
Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:	0.00
The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense	0.00
The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields:	0.00

Names of Personnel receiving masters degrees

NAME
Saad Khan
Fahd Rafi
Ryan Faircloth
Total Number: 3

Names of personnel receiving PHDs

NAME
Yaser Sheikh
Total Number: 1

Names of other research staff

NAME	PERCENT SUPPORTED	
Abhijit Mahalanobis	1.00	No
FTE Equivalent:	**1.00**	
Total Number:	**1**	

Sub Contractors (DD882)

Inventions (DD882)

INTERIM PROGRESS REPORT

ATR Using Multi-view Morphing

Mubarak Shah

Computer Vision Laboratory
University of Central Florida
http://www.cs.ucf.edu/ vision
August 2005

1 Introduction

In this report we briefly detail our preliminary work in reviewing literature and early experimentation for single view morphing. In Section 2, we review current research literature in view morphing and matching for single views, and in Section 3 we discuss a feature based approach to sparse view morphing that also shows promise for compression of the model data corpus. In Section 4 we describe a new Bayesian approach (optimal in a *Maximum a Posteriori* sense) that approaches the problem from a mid-level segmentation problem. We present results of some preliminary experiments for feature-based morphing, as well as preliminary demonstration of the novel object representation.

2 Literature Review

View morphing has been a subject of scientific interest for well over a decade. Here we review some of the recent advancements in the field. Seitz and Dyer's [8] static view morphing algorithm consists of four main steps: determining the fundamental matrix, prewarping, morphing, and postwarping. First, eight or more corresponding points were manually selected to determine a fundamental matrix, $\mathbf{F}$, by using a linear algorithm. Next, the two original images were warped into a plane parallel to the camera baseline using epipolar geometry. Following this, a user manually specified a set of corresponding line segments and then the Beier-Neely method was used to interpolate correspondences. After linearly interpolating the two parallel views based on the disparity map, a parallel morphing view was obtained. To obtain a realistic final view, a quadrilateral was used to determine the postwarping path for the final homography projection using linear interpolation. However, the postwarping path obtained by linear interpolation may cause shrinking problem as mentioned in [8, 10]. Manning and Dyer [5] extended static view morphing to dynamic view morphing. However, in their scenario, the moving objects can only move along a straight line, and their motion is limited to only translation. Xiao et al. [10] relaxed this constraint and allowed an arbitrary motion. They showed that a rigid dynamic scene is equivalent to several static scenes with different epipolar geometries based on relative motion. They also extended

their method to articulated object motion, such as walking and arm gestures, and obtained photo-realistic results. Avidan and Shashua [1] proposed their work on tri-view synthesis by using trifocal tensor. In their framework, an arbitrary novel view can be generated at any 3D view position based on three small baseline images, where the disparity can easily be determined by Lucas-Kanade optical-flow method. It will be almost impossible for Lucas-Kanade method to work for the wide baseline images. Pollard et al. [6] determined edge correspondences and used interpolation to generate a new view over trinocular images. However, since they cannot guarantee that the edge correspondences are correct, their disparity map was computed using the conventional edge-scanline algorithm, which is not clean. Therefore, their results contain a lot of artifacts due to some incorrect correspondences. Vedula et al. [9] proposed view interpolation over spatio-temporal domain, where 1417 fully calibrated cameras (with small baseline) were used on the one side of the actor/actress to capture the events. In their approach, they used voxel coloring, 3D scene flow, and ray-casting algorithm to synthesize the novel view over these original image sequences. They removed the background layer, and only rendered the actor/actress layers. Their results contain some visible artifacts due to the errors in shape estimation, scene flow, etc. Pollefeys and Van Gool [7] combined 3D reconstruction and IBR to render a new view from a sequence of images. They first determined the relative motion between consecutive images, and then recovered the structure of the scene. Next, employing unstructured light field rendering, they can generate a virtual view by using view-dependent texture. Using this sequence of images (small baseline), they accurately estimated dense surface of the scene, which can efficiently improve the visual effect of their results. Recently, Zhang et al. [11] proposed to use feature-based morphing with light fields to obtain very realistic 3D morphing. In their approach, a large number of images (hundreds of pictures) were taken for each object using an array of calibrated cameras. Then, several feature polygons were manually determined employing a user interface. Using the corresponding feature polygons, they generated a 4D light field and grouped the corresponding ray bundles for reference images. Finally, a novel view was synthesized using blending and warping functions on reference images.

3 Sparse (Feature-Based) View Morphing

In order to reason about the continuity of pose change in video, it is important to estimate either the exact pose of some representation of it. In our proposal we described a 2D approach that does not require the extraction of explicit 3D pose, instead uses view morphing to extract a image-based representation of the pose. We achieved success in reconstructing intermediatary views as shown in Figure 2. However, it was evident that manual work is inevitably required for visually plausible view morphing, which in itself is an antithetical to Automatic Target Recognition. Thus, instead of using dense morphing, we propose the use feature based matching instead, firstly since detecting and matching salient features is robust and reliable, and secondly, since it satisfies our requirements of calculating likely poses, which can be leveraged for subsequent multi-view (video) matching. To that end, we use SIFT features [4] for detection and matching. Since the data set observes the object from all points of views, we constrain of

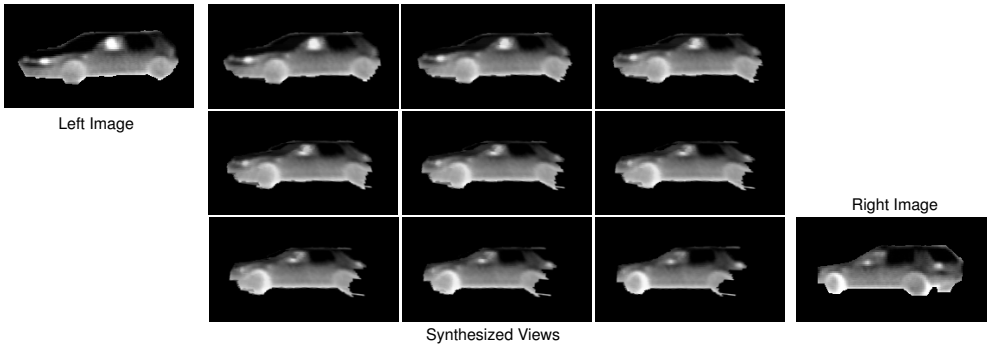


Fig. 1. IR Images and interpolated views.

the movement of any interest point to lie on an ellipse defined by the positions of that interest point in the set of images. Figure 4 shows an interest point (marked by 'o') as it moves along an elliptical curve. In this way we can build a view independent model of the object defined only by the ellipses constraining the positions of certain interest points on the object (or target). Figure 5 shows paths created by the movement of 5 different points on the object as view changes. According to the model, the positions of the interest points plugged into the equation of the defining ellipse should equal zero. The residue can be considered a measure of error. Figure 6 shows residues for one true target and one false set of points. We are looking into estimating such ellipse-based representations for IR images such as Figure 1 as well.

4 Target Recognition

In this section, we describe the object segmentation approach. Rather than necessarily treat images as a collection of pixels arranged on a regular lattice, we assume a more general abstraction of an image as a set of segments on a non-regular lattice. We will describe the overall Bayesian object segmentation framework first, followed by a detailing of training and testing steps.

4.1 Bayesian Framework

By Bayes Theorem,

$$p(\mathcal{L}|\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_n) = \frac{p(\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_n|\mathcal{L})p(\mathcal{L})}{p(\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_n)} \tag{1}$$

Assuming second-order dependency between adjacent segment, the likelihood term can be re-written as,

$$p(\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_n|\mathcal{L}) = \prod_i \prod_j p(\mathbf{x}_i, \mathbf{x}_j|\mathcal{L}) = \prod_i \prod_j p(\mathbf{x}_i|\mathbf{x}_j, \mathcal{L})p(\mathbf{x}_j|\mathcal{L}) \tag{2}$$

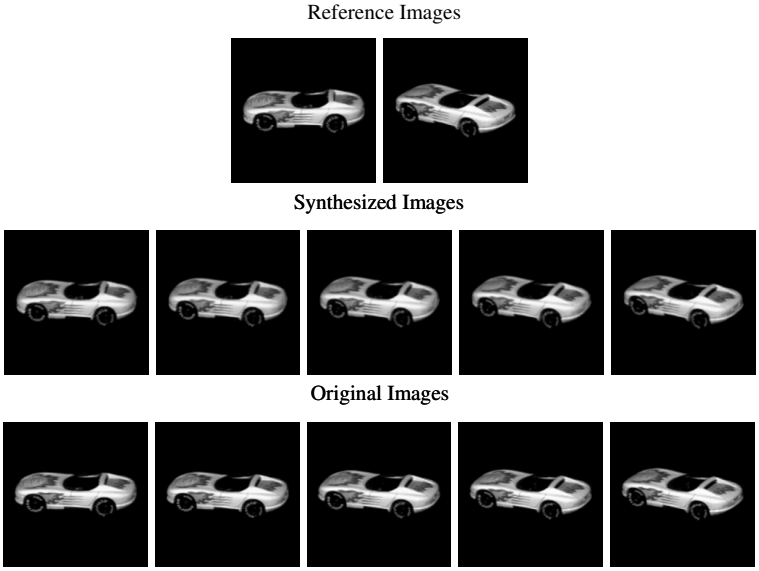


Fig. 2. View morphing from the COIL Data Set.

If a smoothness prior is used, it can be enforced in the decision through a pairwise interaction MRF prior. In particular, the Ising Model is attractive for its discontinuity preserving properties,

$$p(\mathcal{L}) \propto \exp\Big(\sum_{i=1}^{p}\sum_{j=1}^{p}\lambda\Big(\ell_i\ell_j + (1-\ell_i)(1-\ell_j)\Big)\Big), \tag{3}$$

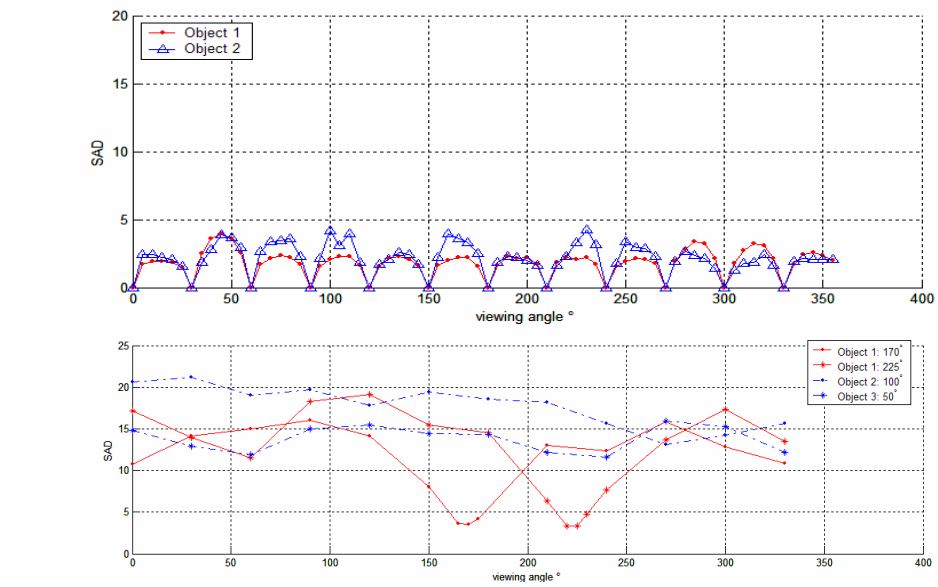
where λ is a positive constant and $i \neq j$ are neighbors as defined in the region-adjacency graph. Ignoring constant terms, that do not affect the optimization, the posterior term is then,

$$p(\mathcal{L}|\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_n) \propto \prod_i \prod_j p(\mathbf{x}_i|\mathbf{x}_j, \mathcal{L}) p(\mathbf{x}_j|\mathcal{L}) \exp\Big(\sum_{i=1}^{p}\sum_{j=1}^{p}\lambda\Big(\ell_i\ell_j + (1-\ell_i)(1-\ell_j)\Big)\Big) \tag{4}$$

Taking the log of both sides and collecting terms,

$$\log p(\mathcal{L}|\mathbf{x}_1, ...\mathbf{x}_n) \propto \sum_i \sum_j \left(\log p(\mathbf{x}_i|\mathbf{x}_j, \mathcal{L}) + \lambda\Big(\ell_i\ell_j + (1-\ell_i)(1-\ell_j)\Big)\right) + \sum_j \log p(\mathbf{x}_j|\mathcal{L}). \tag{5}$$

The MAP estimate is the binary image that maximizes L and since there are 2^{NM} possible configurations of $\mathcal{L}$ an exhaustive search is usually infeasible. In fact, it is known that minimizing discontinuity-preserving energy functions in general is NP-Hard. Although, various strategies have been proposed to minimize such functions, e.g. Iterated Condition Modes or Simulated Annealing, the solutions are usually computationally



(a) (b)

Fig. 3. Several ATR results using view morphing database in COIL. The red lines correspond to two queried images from the same object which can be correcly identified and the viewing angles are also estimated. The blue dot lines correspond to two images from the different objects 2 and 3 whose SADs do not coverge to a small value.

expensive to obtain and of poor quality. Fortunately, since L belongs to the $\mathcal{F}^2$ class of energy functions, a sum of function of up to two binary variables at a time,

$$E(x_1, \ldots x_n) = \sum_i E^i(x_i) + \sum_{i,j} E^{(i,j)}(x_i, x_j), \tag{6}$$

and since it satisfies the regularity condition of the so-called $\mathcal{F}^2$ theorem, efficient algorithms exist for the optimization of L by finding the minimum cut of a capacitated graph. To maximize the energy function, we construct a graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ with a 4-neighborhood system $\mathcal{N}$. In the graph, there are two distinct terminals s and t, the sink and the source, and n nodes corresponding to each image pixel location, thus $\mathcal{V} = \{v_1, v_2, \cdots, v_n, s, t\}$. A solution is a two-set *partition*, $\mathcal{U} = \{s\} \cup \{i | \ell_i = 1\}$ and $\mathcal{W} = \{t\} \cup \{i | \ell_i = 0\}$. The graph construction is with a directed edge (s, i) from s to node i with a weight $w_{(s,i)} = \tau_i$ (the log-likelihood ratio), if $\tau_i > 0$, otherwise a directed edge (i, t) is added between node i and the sink t with a weight $w_{(i,t)} = -\tau_i$. Undirected edges of weight $w_{(i,j)} = \lambda$ are added if the corresponding pixels are neighbors as defined in $\mathcal{N}$ (in our case if j is within the 4-neighborhood clique of i) . The capacity of the graph is $C(\mathcal{L}) = \sum_i \sum_j w_{(i,j)}$, and a cut defined as the set of edges with a vertex in $\mathcal{U}$ and a vertex in $\mathcal{W}$. The minimum cut corresponds to the maximum flow, thus maximizing $L(\mathcal{L}|\hat{\mathbf{x}})$ is equivalent to finding the minimum cut. The minimum

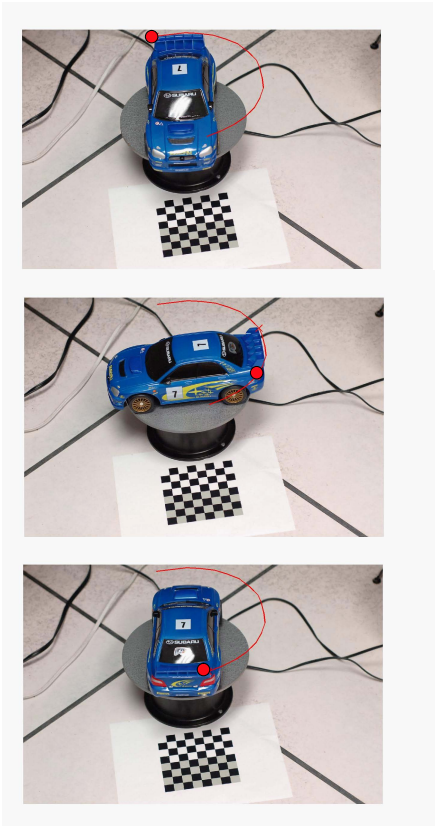


Fig. 4. Corresponding points lie on a conic (assuming an orthographic camera).

cut of the graph can be computed through a variety of approaches, the Ford-Fulkerson algorithm or a faster version proposed by Greig et al. The configuration found thus corresponds to an optimal estimate of $\mathcal{L}$.

4.2 Data Model

The training data for each object is first segmented using mean-shift segmentation [3], which is a non-parametric clustering approach which abstracts each segment as pixels with a common mode in a feature space ($[l\ u\ v\ x\ y]$ is popular choice). From these segments, a region adjacency graph is produced from the segments which is then used first for training and then subsequently for recognition. To evaluate $p(\mathbf{x}_i|\mathcal{L})$, kernel density estimation on the $[l\ u\ v]$ is employed, where each positive segment's mode providing a data point in the feature space. To evaluate $p(\mathbf{x}_i|\mathbf{x}_j, \mathcal{L})$, we create a joint feature space $[l^{(i)}\ u^{(i)}\ v^{(i)}\ l^{(j)}\ u^{(j)}\ v^{(j)}]$ which we populate with every pair of adjacent regions in the training region adjacency graphs.

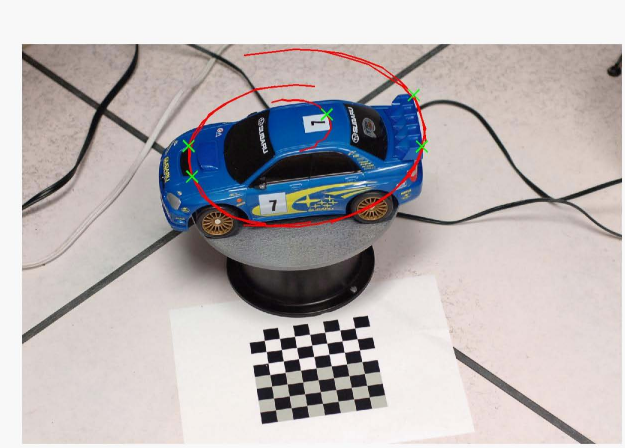


Fig. 5. Paths created by the movement of 5 different points on the object as view changes.

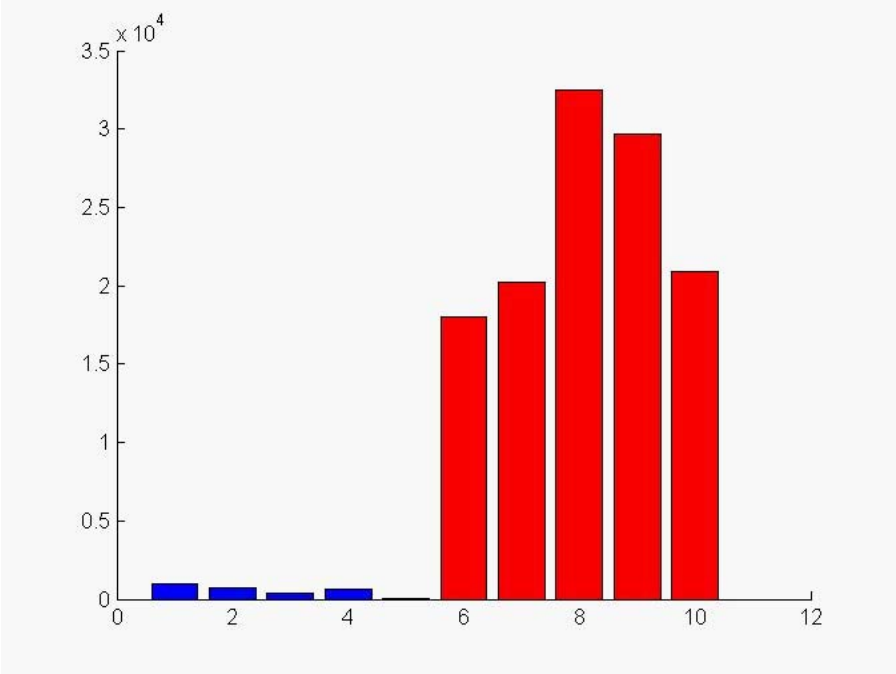


Fig. 6. Residues for one true target and one false set of points.

References

1. S. Avidan and A. Shashua, "Novel View Synthesis by Cascading Trilinear Tensors," *IEEE Transactions on Visualization and Computer Graphics*, 1998.

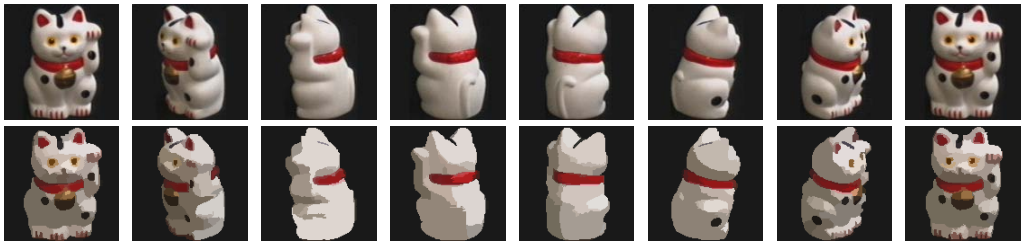


Fig. 7. Several Images from the COIL data set and corresponding segmentations of each image.

Fig. 8. The image, it's segmentationa and the Region Adjacency Graph (RAG) associated with it.

2. T. Beier and S. Neely, "Feature-Based Image Metamorphosis," *ACM SIGGRAPH*, 1992.
3. D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach towards Feature Space Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.
4. D. Lowe, "Distinctive Image Features from scale-invariant keypoints," International Journal of Computer Vision, 2004.
5. R. Manning and C. Dyer, "Interpolating view and scene motion by dynamic view morphing," IEEE International Conference on Computer Vision and Pattern Recognition, 1999.
6. S. Pollard, M. Pilu, S. Hayes and A. Lorusso, "View synthesis by edge matching and transfer," IEEE Workhop on Applications of Computer Vision, 1998.
7. M. Pollefeys and L. Van Gool, "Visual modeling: from images to images," *Journal of Visualization and Computer Animation*, 2002.
8. S. Seitz and C. Dyer, "View morphing," *ACM SIGGRAPH*, 1996.
9. S. Vedula, S. Baker and T. Kanade, "Spatio-Temporal View Interpolation." *Eurographics Workshop on Rendering*, 2002.
10. J. Xiao, C. Rao and M. Shah, "View interpolation for dynamic scenes," *Proceedings of EUROGRAPHICS*, 2002.
11. Z. Zhang, L. Wang, B. Guo and H. Shum, "Feature-based light field morphing," *ACM SIGGRAPH*, 2002.

INTERIM PROGRESS REPORT

ATR Using Multi-View Morphing

Mubarak Shah

Computer Vision Laboratory
University of Central Florida
Http://www.cs.ucf.edu/vision
August 2006

1 Introduction

In this report, we present our works in automatic target recognition using video, which is one of the major tasks of the proposal. In Section 2, we first introduce a novel approach for adaptive video registration. In this approach, the robust layers from a mission video sequence are automatically extracted and a layer mosaic is generated for each layer, where the relative transformation parameters between consecutive frames are estimated. Then, we formulate the image-registration problem as a region-partitioning problem, where the overlapping regions between two images are partitioned into supporting and nonsupporting (or outlier) regions. To determine the corresponding motion parameters, we estimate a set of sparse, robust correspondences between the first frame and reference image. Starting from corresponding seed patches, the aligned areas are expanded to the complete overlapping areas for each layer using a graph-cut algorithm with level set. Next, we estimate the transformation parameters from the mosaic and align the remaining frames in the video to the reference image. Finally, using the same partitioning framework, the registration is further refined by adjusting the aligned areas and removing outliers.

In Section 3, we describe a approach for video-based object recognition which doesn't require any knowledge of camera position or physical location of images with respect to each other. This approach involves a sparse 2D model and object matching on the basis of video. The model is generated based on geometry and image measurements only. We first identify the underlying topological structure of an image dataset and represent it as a neighborhood graph. The graph is then refined by identifying redundant images and removing them using view morphing. This gives a smaller dataset leading to reduced space requirements and faster matching. Finally we exploit motion continuity in video and extend our algorithm to perform matching based on video input and demonstrate that the results obtained using a video sequence are much robust than using a single image. Preliminary experimental results for both approaches are presented and discussed.

2 Video Registration

Image registration and alignment have been studied for a long time in different areas, including photogrammetry, remote sensing, image processing, computer graphics, medical imaging, and computer vision [1–3]. Registration techniques can be classified based on the following two factors: the motion model between mission and reference images, and the method of alignment [2].

The motion model depends on the geometry of the imaged scene and dynamics of the sensor and object motion. Given two images of a planar scene, a single motion model (affine or projective) can be fitted using the existing registration approaches (Fig. 1(a)). For a scene containing multiple planes (or layers), it is difficult to obtain correct registration using only two images (mission and reference) due to the inconsistent motion model. Hence, the registration may overfit one layer or the layer boundaries may not be accurate [4]. However, given a video sequence, an accurate layer segmentation can be obtained by exploiting spatiotemporal information [5–8] (Fig. 1(b)), which makes it possible to perform the layer-based registration.

Alignment methods can be broadly categorized into three classes: intensity-based (or appearance) methods, feature based methods, and hybrid methods. The intensity-based methods are based on the well-known optical flow constraint equation [9], which can be solved by minimizing the sum of squares of pixelwise differences (SSD). Generally, these methods are more useful for frame-to-frame registration of video frames

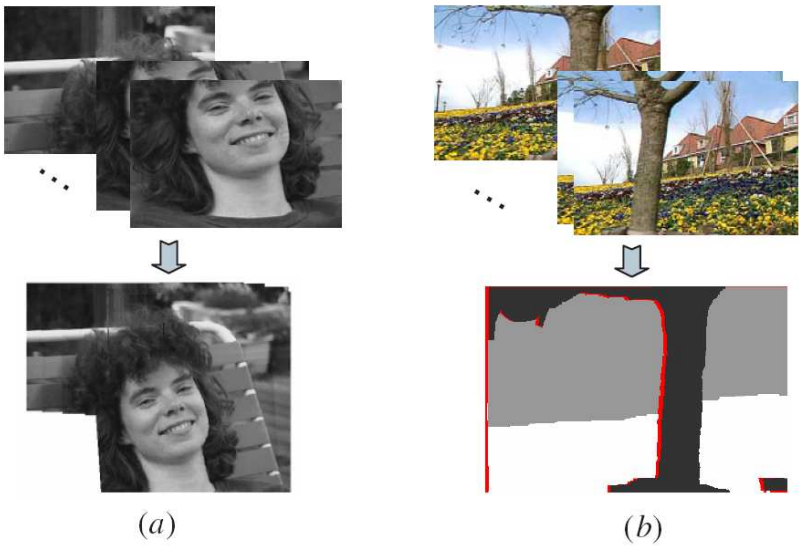
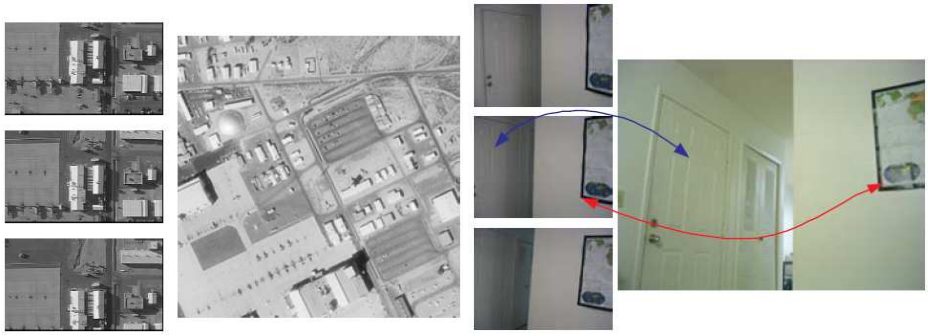


Fig. 1. Depending on the scene, a video sequence can be represented by one layer (a) or multiple layers (b). (**a**) This scene can be approximated by one plane due to the nonparallax camera motion and the generated mosaic. (**b**) "Flower garden" sequence, which can be represented by three layers: tree, garden, and background.

with a simple camera motion, where the pixel motion is small and the image intensities are similar [10, 11]. In the feature-based methods, the main steps include: finding robust features, establishing correspondences, fitting some transformation, and applying the transformation to warp the images [12, 13]. These methods are relatively fast and more suitable for the registration of two dissimilar images with a large and complicated motion or transformation. Recently, several hybrid methods have been proposed to integrate the merits of intensity-based and feature-based methods [14, 15]. In these methods, a set of features is extracted, then an iterative optimization procedure is applied to the supporting regions around these features to minimize some dissimilar measurements.

Currently, some registration problems, such as video mosaicing and registration of video acquired by an airborne sensor to a reference image in the presence of camera information [14, 16, 17], have been solved quite well. However, some problems in this area remain unresolved. First, how do we obtain a reliable initial estimation of motion parameters if the camera information (e.g. location, viewing angles, and sensor model) is not available? Particularly, if camera location and orientation are quite different, such as wide baseline images, the initial estimation usually is quite difficult. Second, how do we deal with outlier regions when the images are taken at different times? These regions may look different due to appearing and disappearing objects, such as moving objects, shadows, and vegetation. Therefore, only a part of the image may be useful for the registration. Third, How do we handle complex motion models in a single 3D scene, such as multiple homographes shown in Fig. 1(b)? Most existing approaches ignore these problems and attempt to align the whole image using a single motion model regardless of the number of layers.

With the aim of addressing the above limitations of the current methods, we propose a novel framework to perform video registration of a 3D scene, which can be approximated by multiple planes, without any knowledge of the metadata. In particular, given an image sequence of a mission or inspection video, we want to register it to a reference image, which may be taken at a different time, location and orientation. The proposed approach first uses a motion layer extraction algorithm [8] to obtain an accurate layer segmentation of the mission video by exploiting spatiotemporal information. For each layer, a mosaic is generated and the relative transformation parameters between consecutive frames are estimated. Then, we formulate the image registration problem into a partitioning framework, where the overlapping regions between two images are partitioned into supporting and non-supporting regions for the registration. In this framework, a region expansion process is designed to adaptively propagate the alignment process from the high confidence seed regions to the low confidence areas and simultaneously remove outlier regions. In order to obtain such starting seed regions, we apply a wide baseline algorithm [18] to compute a set of reliable seed correspondences



(a) (b)

Fig. 2. (a) Three frames from a mission video are shown on the left that are to be registered to a single layer in the reference image shown on the right. (b) Three frames from a mission video are shown on the left that are to be registered to two layers in the reference image shown on the right.

between the first mission frame and reference image. Then, starting from the seed regions, the initially aligned areas are expanded to the whole overlapping areas using a graph cut algorithm integrated with the level set representation of the previous regions. Consequently, we achieve a robust layer alignment for each layer using the relative motion parameters estimated by the layer mosaic, and the final multi-layer video registration is obtained after back projection of layers.

2.1 Single Layer Registration

In a planar scene, only one layer is available, as shown in Fig. 2(a). It is easy to generate a mosaic for this layer using an affine or projective motion model. However, if the scene contains multiple layers, the motion models can vary from a simple global motion model to multiple motion models, where pixel motions are mapped to several parameter clusters. Figure 2(b) shows one example of this case from a "door-wall" sequence, which contains two layers. It is impossible to obtain one mosaic using this mission video without severe misalignment or distortion. Fortunately, in the context of video registration, temporal information is available in the mission video sequence, from which the motion layers of the scene can effectively be extracted. In this paper, we use a multiframe graph-cut framework [8] to achieve an accurate layer segmentation of the mission video sequence. After the motion segmentation, we obtain precise supporting regions for each layer and the corresponding motion parameters between each consecutive frame, which can be used as the initial parameters for layer mosaicing.

Since the gap between consecutive frames of a video sequence is small, it is better to use an intensity-based registration method to minimize the image residue (or SSD), which can be written as

$$\epsilon = \sum_{\Omega} [I_2(H \cdot \mathbf{x}) - I_1(\mathbf{x})]^2, \tag{1}$$

where I_1 and I_2 are two original images, Ω is the overlapping area between two consecutive frames, $\mathbf{x} \in \Re^2$ are the homogenous coordinates and $H \in \Re^{3\times 3}$ is a homography matrix between two frames. Starting with $H = \mathbf{I}$ (identity matrix), a nonlinear approach, such as LevenbergCMarquardt method, can be used to iteratively minimize the residue [11]. In this method, after computing image gradient ∇I from the two images, a gradient-descent direction is estimated that leads to a local minimum.

The transformation H_i between mission frame i with the mosaic becomes known after a mosaic is generated for each layer. Therefore, we have two choices when it comes to aligning the mission video to the reference image as shown in Fig. 3. In the first scheme, after aligning the layer mosaic to the reference image with transformation F, an initial transformation for a mission frame i to the reference image can be computed by $T_i = FH_i$. However, in this scheme, the error between frame f_i with frame f_1 will be accumulated with i increasing, which may not provide a good estimation between the layer mosaic and the reference image.

In our work, we use an alternative solution, whereby each frame f_i is directly registered to the reference image based on the previous transformation of frame f_{i-1}. First, we align the first frame to the reference

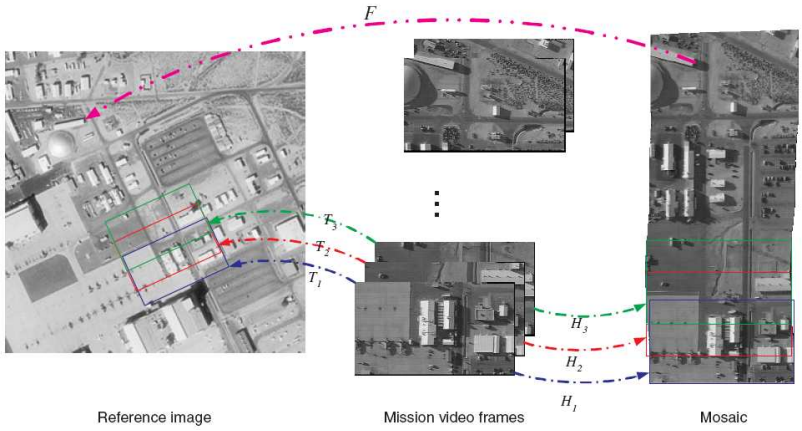


Fig. 3. The transformation among mission frames, reference image, and layer mosaic for one georegistration sequence. H_i is the transformation between mission frame i with mosaic. F is the transformation between mosaic and reference image. T_i is the final transformation between mission frame i and reference image.

image and estimate its corresponding transformation T_1 by determining corresponding seed regions and using the region expansion approach. Then the initial transformation for the second frame can be simply computed by $T_2 = T_1 H_1^{-1} H_2$, which can be further refined only using the region expansion process without computing seed correspondences. After estimating the precise transformation T_{i-1} of f_{i-1}, we iteratively compute the initial transformation for frame i by $T_i = T_{i-1} H_{i-1}^{-1} H_i$ using the previous frame f_{i-1}. As a result, we can avoid the accumulated error of the mosaic since the initial transformation is always computed employing the previous frame f_{i-1} instead of the first frame f_1. Hence, before registering the whole mission video sequence, we have to align frame f_1 to the reference image and compute T_1.

2.2 Layer Registration

The first issue that has to be tackled for layer registration algorithm development is the decision to use either sparse or dense image features for registration. Given two wide-baseline images without any metadata, it is difficult to perform alignment due to illumination variations and large motion between two images. Therefore, the use of sparse image features is ideal for the fast estimation of initial motion parameters. However, due to outliers and inaccuracy of these correspondences, the initial registration is usually not good enough. In this section, we propose a two-stage approach to integrating the merits of the sparse and dense image features. In the first stage, we determine a set of sparse correspondences between the mission and reference images. Then, starting from the initial seed correspondences, the aligned regions are gradually expanded to cover the whole overlapping areas between both images. At the same time, the outlier regions, such as appearing/disappearing objects that may harm the registration process, are detected and removed.

There are several methods for computing robust correspondences for wide-baseline images [19, 20]. Here we make use of our previous algorithm [20] to determine a set of reliable corresponding corners. In this approach, a set of edge-corners is identified in both images that provide robust and consistent matching primitives. Figure 4 shows the detailed matching process for a small patch. After computing the best affine transformation and illumination coefficients between the two patches in mission and reference images, the patch in I_2 (Fig. 4(d)) is warped as shown in Fig. 4(e), which has a similar appearance to the patch in I_1 (Fig. 4(c)), and the residue is minimized.

Once the seed correspondences are estimated between the mission and reference images, our purpose is to perform registration for each plane (layer) in the scene. For each pair of correspondences, we consider a small patch centered around each seed region, which can be approximated as a planar patch (or an initial layer) in the scene. Therefore, we get a number of initial layers, and each layer is supported by a small square region with its corresponding affine transformation. This implies that the corresponding small square regions in the mission and reference images are aligned by this initial affine transformation.

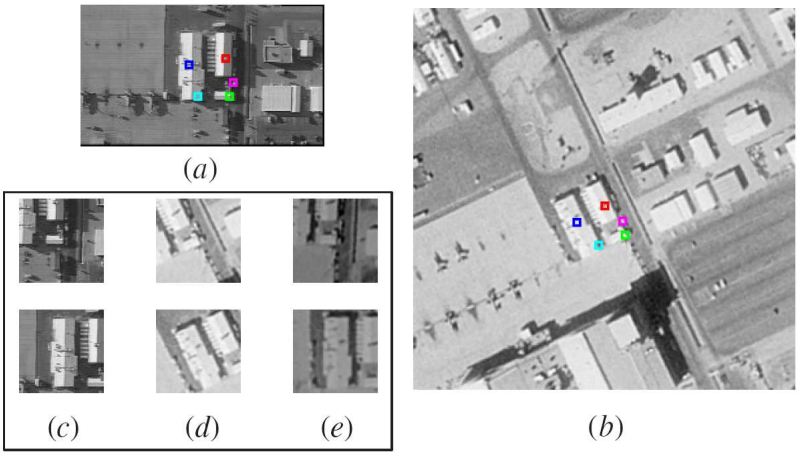


Fig. 4. Determining correspondences between the mission frame and reference image. (**a**) Mission image. (**b**) Small part of reference image. Several correspondences are computed by the wide-baseline matching algorithm, each pair of correspondences is marked by squares with the same color. (**c**)-(**c**) Matching process of green (top row) and blue (bottom row) corners. (**c**) A patch from (**a**). (**d**) Corresponding patch from (**b**). (**e**) Warped patch (**d**) obtained after applying the best affine transformation, where patch (**e**) is similar to patch (**c**). NB: Compared to the original patches (**c**) and (**d**), the illumination effect is partially compensated between (**c**) and (**e**) by estimating μ and δ.

Nevertheless, this minimization process may create two problems. First, the estimated parameters obtained by using the small patch may overfit the pixels inside the region and may not correctly represent the global transformation of a larger region. Second, this process ignores the appearing/disappearing objects between two images, such as the moving objects, occlusion areas, and shadows. To overcome the problems described above, we expand the region boundary to obtain more supporting pixels that are consistent with the motion parameters and also to identify the outlier pixels. Then we iteratively refine the motion parameters using these supporting pixels. Therefore, this registration problem is essentially converted into a partitioning problem that can be stated as follows: Determine the optimal supporting regions and their corresponding motion parameters for image registration.

Our registration problem can be recast into the graph-cut framework. In this framework [1], we seek the labeling function f that partitions the pixels in region Ω into two groups: the first group represents the supporting regions, labeled $f = 0$; the other represents the outlier regions, labeled $f = 1$. This partitioning can be achieved by minimizing the following energy function:

$$E = \sum_{(p,\, q)\in N} V(p,\, q) + \sum_{p\in\Omega} D_p(f_p) \tag{2}$$

where the first term is a piecewise smoothness term, the second term is a data penalty term, N is a 4-neighbor system, and f_p is the label of a pixel p. $D_p(f_p)$ can be approximated by a Heaviside function.

To minimize the energy function, a weighted graph $G = \langle V, E\rangle$ is constructed, where V is a node set (image pixels) and E is a link set that connects the nodes. After assigning weights for the links, we can compute a minimum cut $\mathcal{C}$ using a standard graph-cut algorithm and partition the original region into the supporting and outlier regions. However, using this process we cannot expand the region from the initial seed patch to the exterior to obtain more supporting pixels. Hence, we must use the contour of the previous seed region prior to computing the level set representation for this region [21,22], which allows the region contour to evolve along the normal direction. After enforcing the level set regulation on the sink-side weight of graph $\mathcal{G}$, we can effectively control the graph-cut algorithm to gradually expand the seed region.

Figure 5 shows a detailed expansion process starting from one initial seed region. Figures 5(a) and (b) show the initial contours of the corresponding seed regions. Based on the initial contour of the original seed region Ω^0 (Fig. 5(b)), we construct a mask β of this region, which has a value in $[0, 1]$, where the interior pixels of the region are marked by 1 and the others are marked by 0. Then, a level set ϕ (Fig. 5(e)) can be simply computed by convolving the region mask with a Gaussian kernel as: $\phi = G * \beta$, where the value

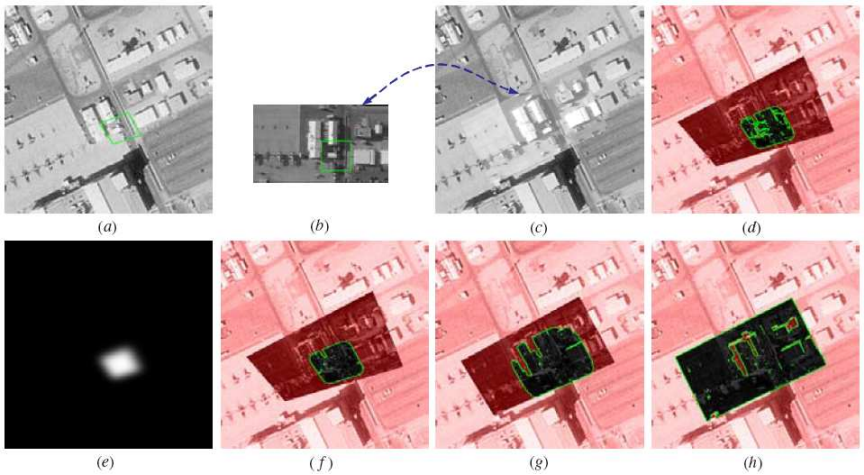


Fig. 5. Region expansion process. (**a**), (**b**) Initial corresponding patch contours in the reference and mission images, respectively. (**c**) Final registration result, where the intensities of the embedded mission image are adjusted by illumination coefficients μ and δ. (**d**) Simple expansion and partitioning started from the initial contour shown in (**a**). (**e**) Level set representation of initial contour (**a**). (**f**)-(**h**) Intermediate results using graph-cut method with the level set representation, which can guarantee that the expansion gradually evolves from the center to a boundary. NB: The green boxes in (**a**) and (**b**) are the initial seed regions. (**f**)-(**h**) Difference images between the warped (**b**) and (**a**) and the green contours in (**f**)-(**h**) are supporting region boundaries obtained after using a bipartitioning algorithm. The nonsupporting pixels are masked by red.

of ϕ falls down along the contour normal direction until $\phi_p = 0$. Then, we warp the second image using the corresponding homography and construct a graph $\mathcal{G}$ for the pixel with $\phi_p > 0$. After that, we apply the level set ϕ to change the weight of the sink-side*t*-link for each pixel, such that the weights of the pixels inside the region are almost unchanged while the weight $(p,\ t)$ will decrease when the pixel p is away from the boundary. As a result, the minimum cut $\mathcal{C}$ is most likely to exclude the outside pixels and label them as the non-supporting pixels for this region. This way, the new expanded supporting region Ω^1 can be computed as shown in Fig. 5(f). After several iterations as shown in (Figs. 5f-h), the region's boundary gradually propagates from the center to the exterior until it reaches the overlapping boundary of two images, and the alignment is stable. Figure 5(h) shows the final region Ω^5 after five iterations, and Fig. 5(c) shows the final registration results using the projective transformation computed by this approach. If several initial seed regions share the same motion transformation for some layer, we expand the multiple regions simultaneously to speed up the registration process. Figure 5 shows that our approach can obtain the piecewise smooth region expansion, which is insensitive to noise. The outlier regions due to shadows are also detected and removed. At the same time, the transformation T_1 for the key frame is estimated. After applying the initial transformation $T_i = T_{i-1}H_{i-1}^{-1}H_i$ to frame f_i, we initialize the alignment of frame i to the reference image. Then, employing the region expansion approach to the ith frame, we remove outliers and refine the alignment to compute the transformation T_i for this frame. The final video registration results are shown in Fig. 6.

2.3 Experiments

We performed several experiments on different real data sets, where the metadata information was not available. In all of the experiments, we applied the wide-baseline matching algorithm to estimate sparse correspondences, which can provide an approximated initial alignment between mission and reference images. For a single layer registration, after determining the sparse correspondences, we expanded these seed regions simultaneously to speed up the alignment process. The initial homography between two images could be computed in two ways: select the most robust affine transformation of the seed regions using the RANSAC technique or estimate a homography voted on by all of these correspondences.

In Fig. 7, we show an example of the multi-seed expansion process. Since a number of correspondences are determined, it is easy to estimate a robust initial homography using all the correspondences. Then, starting

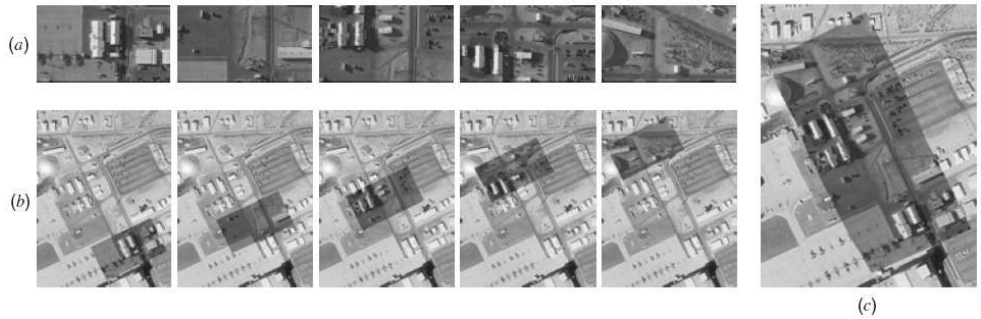


Fig. 6. Video registration results. (**a**) Mission video frames. (**b**) Registration results for several frames, where the mission images are superimposed on the reference image. (**c**) Full registration of all mission video frames.

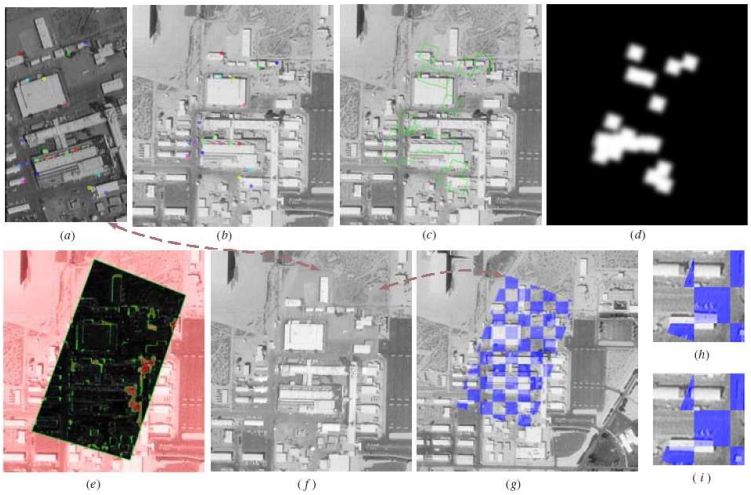


Fig. 7. Registration using multiseed region expansion. (**a**) Mission image. (**b**) Small part of a reference image. The correspondences are marked in a and b by the same colors. (**c**) Initial seed regions and (**d**) corresponding level set representation. (**e**) Final region contour after expansion, where the nonsupporting regions are indicated by red. (**f**) Registration results. (**g**) Checkered display after alignment. (**h**), (**i**) Zoomed alignment results before and after applying the region expansion alignment.

with the initial homography, we expand all the initial seed regions simultaneously until the overlapping areas between the mission and reference images are covered. Our graph-cut algorithm also detects and removes the outlier regions, most of which are due to vegetation or shadows. Figures 7(h) and (i) compare the zoomed results before and after applying the region expansion process.

Figure 8 shows another set of results for geo-registration using single seed region expansion where only three correspondences are determined due to the small size of the mission frame. Since we cannot obtain a good initial projective transformations from these few correspondences, we use RANSAC to determine the robust affine transformation of the seed regions, which is shown in blue in Figs. 8(a) and (b). Then, starting from one seed region (blue), we perform the adaptive region expansion alignment and obtain the registration results as shown in Figs. 8(c) and (d).

Figure 9 shows the final registration results for the doorwall sequence. After obtaining the layers for each frame, we align the different layers to the reference image separately using the adaptive region expansion approach. The final registration results of the first frame are shown in Figs. 9(d) and (e). Compared to the direct registration, our approach has two advantages. First, to align the corresponding layers, we employ different sets of motion parameters to correctly represent the mapping of the pixels in these layers. Second, the layer segmentation also provides accurate supporting regions for each layer, which prevent the region expansion process across the layer boundaries. Therefore, for each layer registration, our approach can effectively avoid

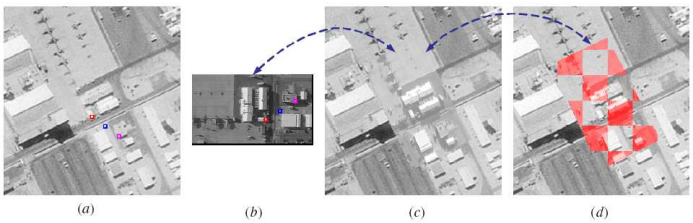


Fig. 8. Registration using one seed region expansion. (**a**) Small part of a reference image. (**b**) Mission image. (**c**) Registration results. (**d**) Checkered display after alignment.

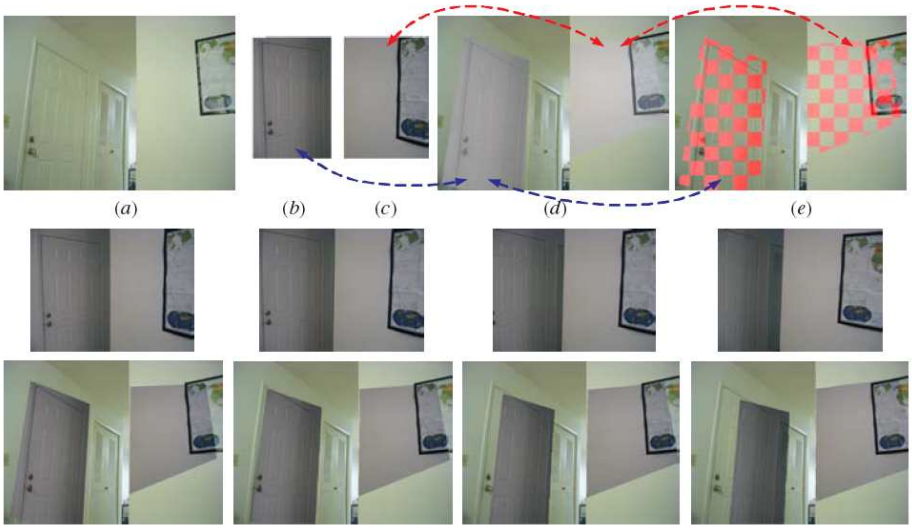


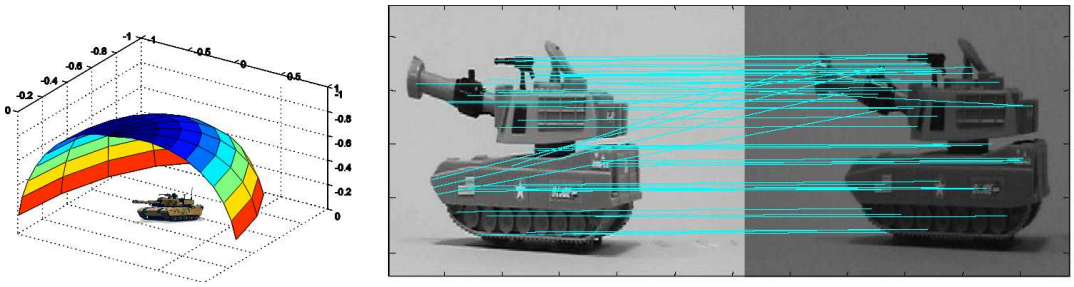
Fig. 9. Multiple-layer registration. (**a**) Reference image. (**b**), (**c**) Layers of door and wall in frame 1. (**d**) Registration results of frame 1. (**e**) Checkered display after alignment. Middle: Some mission video frames. Bottom: Corresponding video registration for these frames, where the mission images are split into two parts during the registration.

the pixels from the other layers and achieve more accurate aligned regions for each layer. In all of our experiments, after determining correspondences, the computational time for a single layer registration is less than 10s per frame.

3 Model Generation for Video-based Object Recognition

In this section, we present a strategy for object recognition. This needs techniques to extract information on the basis of the contents of the images without human intervention and identify the objects present in them. What makes this task difficult is the difference of interpretation of images by humans and software systems. Most current systems operate on the low-level features like color or texture, directly extracted from the pixel, while humans go for the semantic meaning or the high-level features present in the image, like the objects or scene it contains [23]. Hence, one of the major tasks in image analysis is to identify the different objects present in the scene. Another major difficulty arises due to the fact that the operating conditions may differ significantly from those of training and are not anticipated [24]. The major issues and needs of object recognition include good representation of object models and backgrounds, adaptation to facet or environment changes, good features for object representation and efficient use of a priori knowledge about object signatures [25].

There are a variety of approaches explored for object recognition, like, CAD-based, appearance-based and shape-based methods; however, each approach has its own set of limitations [26]. In each of the techniques, a model is generated which is then compared with an image to identify the object being tested. However,



(a) (b)

Fig. 10. (a) Tessellating the viewing space. (b)Matching of Feature points.

object recognition from a single view may fail when there is much similarity among test objects or when the background clutter or partial occlusion masks features of the object. Selinger *et. al* [27] used multiple fixed cameras of known pose to apply single view object recognition system over a sequence of imagery. However, they did not find any significant advantage of this approach. Later on, Zhou *et. al* [28] successfully utilized the temporal information present in video sequences for face recognition. They formulated a probabilistic model merging the dynamics and identity of humans obtained from video. However, they assumed certain constraints in the motion of persons while gathering their test data. Javed *et. al* [29] presented a probabilistic framework for general object recognition using a video sequence containing different views of an object. They generated a model for each object in the training set capturing images at known viewing angles of camera and poses of objects.

In our approach, we use a set of reference images to generate an online sparse 2D model, estimate the underlying topological structure and, arrange them in the form of a connectivity graph. We refine the graph using morphing, so as to remove the redundant images and finally use video matching for recognition of objects. The strength of our approach is that we don't need to know the object pose beforehand; and the video sequence could be shot over any arbitrary trajectory with objects following an unconstrained (but smooth) path. The use of video rather than a single image increases the confidence measure of the match.

3.1 Model Generation For Objects

Any object can be modeled using either an object-centered or a view-centered representation [30, 31]. The object-centered representations use the features from the objects, like boundary curves, surfaces etc, to describe the volumes of space. View-centered representations, on the other hand depend, on the outlook of objects from different viewpoints. These involve the use of aspect graphs and silhouettes for modeling. We have used the view-centered representation for generation of database, which makes the task of matching simpler. This is because the need for projection of model to 3D is no longer there and the features that are to be compared are in 2D [31]. The input to our database generation algorithm is a set of reference images, which have been arbitrarily extracted from a video sequence shot around an object. Our system tessellates the images around the viewing space of the object, as shown in Fig. 10(a). The algorithm generates a neighborhood graph, where each image is identified as a node and the links between neighbors are specified as edges. The images are defined as neighbors on the basis of their logical proximity and extent to which they match with each other.

Development of Neighborhood Graph Given a set of reference images, we propose a novel approach to tessellate them around the viewing space of the object while ensuring a minimal size of the database. The algorithm begins by identifying the feature points in all the images of the repository. We have used the Scale-Invariant Feature Transform (SIFT) Operator to extract the distinctive features in the image. The features are invariant to image scale and rotation; and robust to changes in viewpoints and illumination. Feature coorespondences are then identified using a fast nearest-neighbor algorithm [32], which are ultimately used to decide the presence or absence of linkage between nodes. Figure 10(b) shows SIFT points and matches identified for a pair of images.

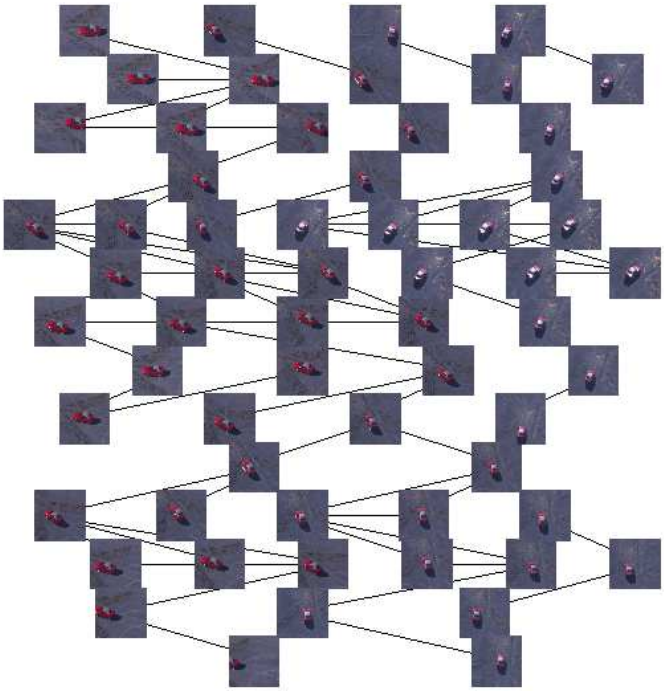


Fig. 11. Neighborhood Graph for a car.

For an image database having originally N images, an $N \times N$ link matrix is formed. A link between image pair (I_i, I_j) is marked if they are found to be neighbor. The procedure for identification of neighbors is two-fold. In the first pass, we find the average Euclidean distance d for each image pair (I_i, I_j). For c corresponding points between two images, we have:

$$d(I_i, I_j) = \frac{\sum_{k=1}^{c} \sqrt{(I_{ik}^x - I_{jk}^x)^2 + (I_{ik}^y - I_{jk}^y)^2}}{c} \tag{3}$$

For each image, the pair with the minimum distance is selected as the neighbor, and an edge is marked between them. Considering this attribute as our seed point, we expand the region to include all those images in the neighborhood block, whose Euclidean distance falls within 25% of the minimum value. This accounts for the out of plane images and handles arbitrary viewpoints.

In the second pass, we apply the physical proximity constraint between successive video frames. This implies that two consecutive frames of a video sequence represent two images in proximity and hence represent neighbors. Therefore:

$$Neighbor(I_i,\ I_{i+1}) = 1, \quad \forall\, i \in \text{Set of Frames} \tag{4}$$

It may be noted that this second criterion improves the connectivity of the graph. In cases, where the image set is not from a true video sequence, and represents an arbitrary collection of images, only the first criterion would suffice. Figure 11 shows a portion of a graph that is generated for a car. Such a graph is generated for each object and stored as a model.

Multi-view Morphing Once the Neighborhood graph is generated, it is refined using view morphing. Seitz *et. al* [33] introduced view morphing to generate novel views from varying viewpoints using only two images. Their approach is based on the principles of projective geometry, which can explicitly preserve 3D information. Given sparse correspondences between the image pair, view morphing works by rectifying the two images in such a manner that the corresponding points lie in the same scanline (a step known as prewarping). This allows calculation of disparity map which helps in retrieving dense correspondences. Once

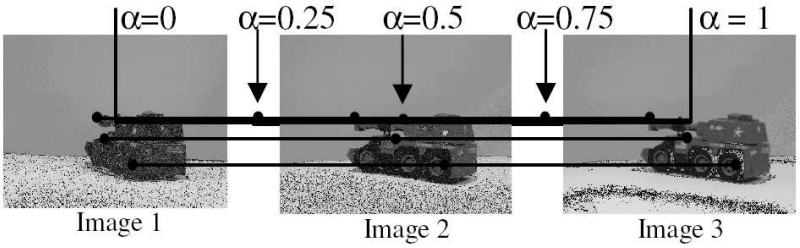


Figure 4. Removal of redundant node through morphing.

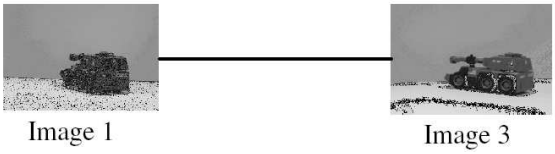


Fig. 12. Updated Neighborhood Graph after Morphing.

the dense correspondences are known, the morph is generated using cross dissolve, and the resulting image is re-projected to its final position. Seitz's work could however be used to generate new views only along the line connecting the two original images. Later on Wexler *et. al* [34] extended the concept to tri-view morphing and were able to synthesize morphs at any viewpoint within the boundaries of the triangle formed by the three images.

Graph Pruning A view-centered approach leads to a space requirement that is larger than that of object-centered representation [31]. This is because many characteristic features are to be noted and there might be an overlap among the images. This requires special attention to be paid to keep the size of database at minimum. We proceed by analyzing for each image if it represents a morph of its neighbors or not. To test any image I_i we begin with extracting its two adjacent images I_j and I_k and apply morphing on them to generate features and verify if they represent the features originally extracted from I_i or not.

Given an Image pair (I_j, I_k), with corresponding feature points p and q, we align the image pairs to have the corresponding points along corresponding scan lines and synthesize the features using Equation (3) for varying values of α:

$$p_s = p\alpha + (1 - \alpha)q \tag{5}$$

The features generated in this manner are compared with the original features extracted from I_i. For this, we have to iteratively engender and compare p_s for varying values of α. If there exists an α for which ps represents the features of Ii, it means I_i could be generated using I_j and I_k and hence could be removed from the dataset. This procedure is demonstrated in Fig. 11 and updated graph is shown in Fig. 12. This proceeds till all the images in the database are exhausted.

The same procedure is then repeated for images having larger number of neighbors. The strength of this technique is that we do not have to generate the intermediate images completely. Rather, we simply work on the selected features of the images. This saves us from computing the disparity map which takes time.

3.2 Video-Based Object Recognition

One way to identify the target image is to generate a massive dataset of virtual views using morphing and compare the test image with all of them. This is inefficient and computationally expensive. We propose to initially match the test image with only those images stored in the database. This helps in identifying an approximate neighborhood of the image being examined. Once a seed image is found, the virtual images around it could be generated using the morphing approach and compared with the test image. Since we do not have to generate the whole image; rather, we work with the sparse features detected by the feature detector, the speed of our system is increased due to elimination of the disparity map generation step.

In order to further strengthen the confidence measure of our detection results, we have used video sequences instead of single image for target recognition. The major advantage of this technique is that the

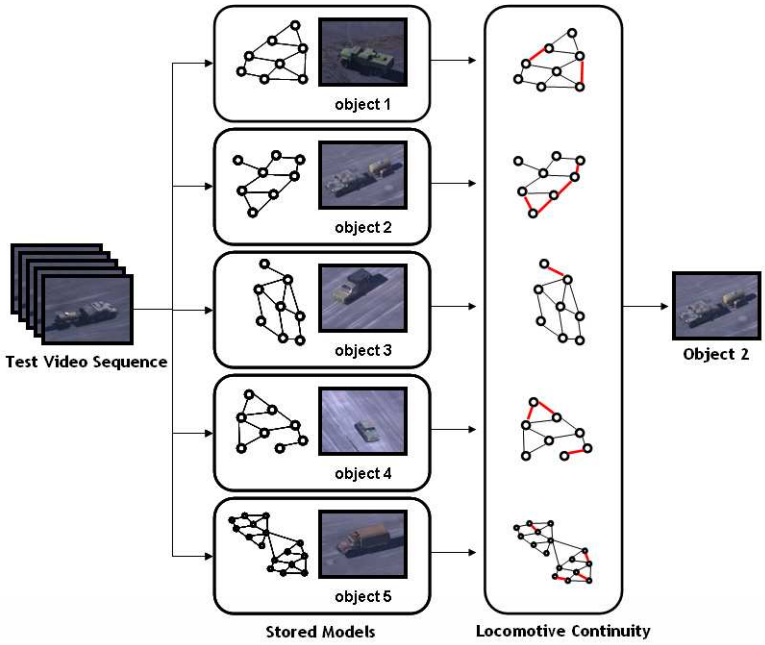


Fig. 13. Using Video Sequence for Matching.

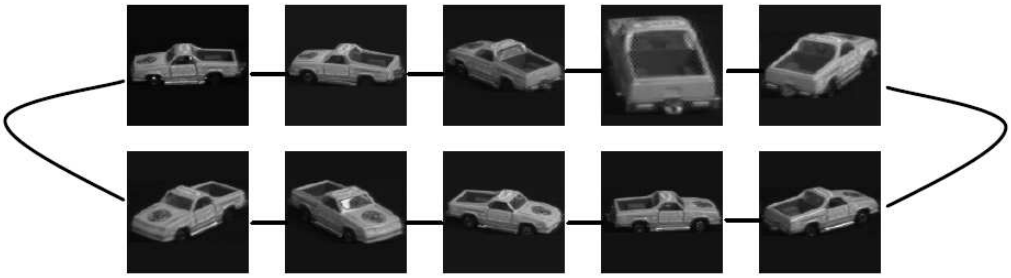
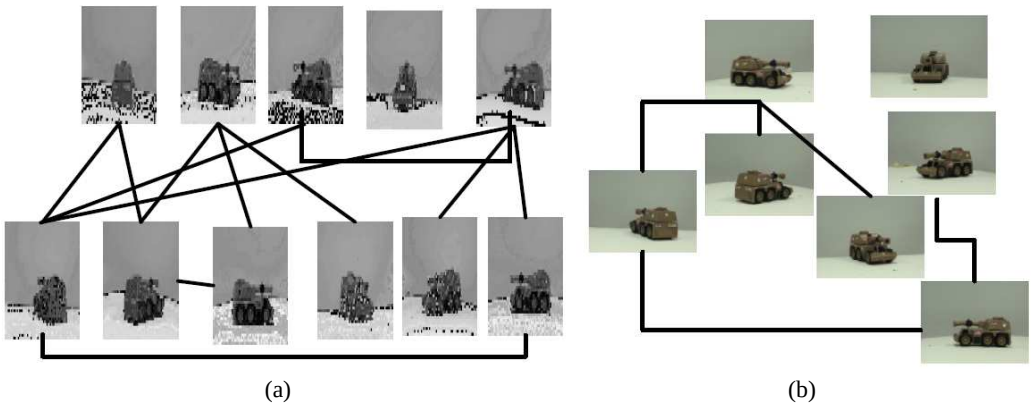


Fig. 14. Linear Graph Generated for an Object of COIL dataset.

video provides information of multiple views. Many objects in real world look alike, if observed from a particular viewpoint and completely different when observed from some other point of reference. Using a video for object recognition, we can exploit the fact that the two adjacent images in the video sequence represent proximally closer views of the object. Hence, the adjacent frames of the video sequence should point to the same (or adjacent) nodes of the neighborhood graph. Thus, a correct identification results in a smooth transition across the multiple images, following an unbroken trajectory in the model. On the other hand, an incorrect match results in jitters across the multiple frames, which helps in identifying the incorrect matches. Our approach for developing the topological structure of the images in database provides ease of traversing while using video sequence. As shown in Fig. 13, given the stored networks of objects and a test video sequence, only the correct object follows a smooth trajectory along the graph and others suffer from discontinuities.

3.3 Experiments

To evaluate our approach for target recognition, we used the Columbia Object Image Library (COIL100) data set from the Columbia University and VIVID by DARPA. In COIL there are 72 images each of 100 objects. The viewing angle between these images is uniform, and this leads to a fairly linear neighborhood graph. See



(a) (b)

Fig. 15. (a) A portion of original neighborhood graph and (b) its pruned Network.

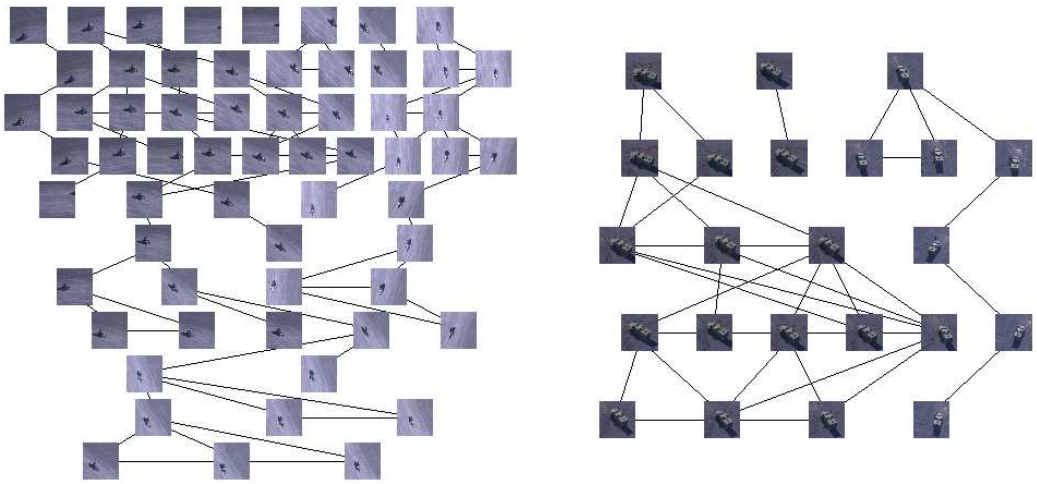
Fig. 14 for neighborhood graph generated for a pickup. In order to capture the randomness of the real-world image capture, we shot our own video sequences following arbitrary trajectories.

Figure 15 shows a portion of the neighborhood graph of one of the objects and the updated Network. Experiments show that our algorithm could generate the neighborhood graph with a precision of 97.86%. The system was able to reduce the image base to about 60% of its original size.

Video based matching improved the results obtained from single image matching. Single image matching gave 40% correct matches, while video-based recognition gave about 80% correct matches. The reason for the 20% incorrect matches is the high similarity of different objects at certain poses, which further increases the viability of our approach of using videos instead of single image for matching. The Fig. 16(a) shows a smooth trajectory for the correct identification of motor bike. Figure 16(b) identifies an incorrect matching of a Humvee with the green truck by pointing discontinuities.

References

1. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. In: Internation Conference on Computer Vision. (1999)
2. Fitzgibbon, A., Zisserman, A.: Automatic camera tracking. In Shah, M., Kumar, R., eds.: Video Registration. Kluwer (2003) 18–35
3. Zitova, B., Flusser, J.: Image registration methods: a survey. **21**(11) (2003) 977–1000
4. Wills, J., Agarwal, S., Belongie, S.: What went where. (2003) 37–44
5. Ayer, S., Sawhney, H.S.: Layered representation of motion video using robust maximum-likelihood estimation of mixture models and mdl encoding. In: Fifth International Conference on Computer Vision, Cambridge, Massachusetts (1995) 777–784
6. Ke, Q., Kanade, T.: A robust subspace approach to layer extraction. In: IEEE Workshop on Motion and Video Computing. (2002) 37–43
7. Khan, S., Shah, M.: Object based segmentation of video using color motion and spatial information. (2001)
8. Xiao, J., Shah, M.: Motion layer extraction in the presence of occlusion using graph cut. In: CVPR. (2004)
9. Horn, B.K., Schunck, B.G.: Determining optical flow. Artificial Intelligence **17** (1981) 185–203
10. Sawhney, H., Hsu, S., Kumar, R.: Robust video mosaicing through topology inference and local to global alignment. (1998) II: 103
11. Szeliski, R.: Video mosaics for virtual environments. IEEE CG&A (1996) 22–30
12. Brown, L.G.: A survey of image registration techniques. ACM Comput. Surv. **24**(4) (1992) 325–376
13. Zheng, Q., Chellappa, R.: A computational vision approach to image registration. T-IP **2** (1993) 311–326
14. Keller, Y., Averbuch, A.: Implicit similarity: a new approach to multi-sensor image registration. (2003) II: 543–548
15. Shen, D., Davatzikos, C.: Hammer: hierarchical attribute matching mechanism for elastic registration. **21**(11) (2002) 1421–1439
16. Sheikh, Y., Shah, M.: Aligning 'dissimilar' images directly (2004)
17. Sheikh, Y., Khan, S., Shah, M., Cannata, R.: Geodetic alignment of aerial video frames. (2003) Chapter 7
18. Wildes, R., Hirvonen, D., Hsu, S., Kumar, R., Lehman, W., Matei, B., Zhao, W.: Video georegistration: Algorithm and quantitative evaluation. (2001) II: 343–350



(a) (b)

Fig. 16. (a) Smooth trajectory for a correct match for motor cycle. (b) Jitters representing an incorrect match for a green truck.

19. Ferrari, V., Tuytelaars, T., Van Gool, L.: Wide-baseline multiple-view correspondences. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Volume 2. (2003) 718–725
20. Xiao, J., Shah, M.: Two-frame wide baseline matching. In: International Conference on Computer Vision. (2003)
21. Osher, S., Fedkiw, R.: Level set methods and dynamic implicit surfaces. Springer (2003) OShE st 03:1 1.Ex.
22. Sethian, J.A.: Level set methods and fast marching methods. Cambridge University Press (2001) Sethian.
23. Hove, L.J.: Extending image retrieval systems with a thesaurus for shapes. Master thesis (2004)
24. Ravichandran, B., Gandhe, A., Smith, R.E.: Xcs for robust automatic target recognition. In: GECCO 2005: Proceedings of the 2005 conference on Genetic and evolutionary computation. Volume 2., ACM Press (2005) 1803–1810
25. Roth, M.W.: Survey of neural network technology for automatic target recognition. In: IEEE Transactions on Neural Networks. Volume 1. (1990) 28–43
26. Xiao, J., Shah, M.: Automatic target recognition using multi-view morphing. In: Proceedings of SPIE on Automatic Target Recognition XIV. (2004) 391–399
27. Selinger, A., Nelson, R.: Appearance-based object recognition using multiple views. (2001) I:905–911
28. Zhou, S., Kruger, V., Chellappa, R.: Face recognition from video: A condensation approach. (2002) 212–217
29. Javed, O., Shah, M., Comaniciu, D.: A probabilistic framework for object recognition in video. (2004) IV: 2713–2716
30. Bennamoun, M., Recognition, G.M.O.: Fundamentals and Case Studies. Springer (2002)
31. Pope, A.R.: Model-based object recognition - a survey of recent research. (1994)
32. Lowe, D.: Distinctive image features from scale-invariant keypoints. **60**(2) (2004) 91–110
33. Seitz, Dyer: View morphing. In: Computer Graphics Proceedings, Annual Conference Series (Proc. SIGGRAPH '96). (1996) 21–30
34. Wexler, Y., Shashua, A.: On the synthesis of dynamic scenes from reference views. (2000) I: 576–581

INTERIM PROGRESS REPORT

ATR Using Multi-View Morphing

Mubarak Shah

Computer Vision Laboratory
University of Central Florida
Http://www.cs.ucf.edu/vision
August 2007

1 Summary

In this report, we present a novel method for object class detection which is based on 3D object modeling. Instead of using a complicated mechanism for relating multiple 2D training views, our method establishes spatial connections between these views by mapping them directly to the surface of 3D model. The 3D shape of an object is reconstructed by using a homographic framework from a set of model views around the object and is represented by a volume consisting of binary slices. Features are computed in each 2D model view and mapped to the 3D shape model using the same homographic framework. Also we present our work on object recognition based on correlation using morphing technique. There has been considerable interest in using correlators for pattern recognition. Correlators are inherently shift-invariant allowing us to locate patterns (such as moving targets) in the input scene merely by locating the correlation peak. Thus, we do not need to segment or register the images prior to correlation, as we have to do in alternate methods for pattern recognition. In this report, we describe two new methods for synthesizing new views of a known object so that the occluded features of the object can be inferred and incorporated into the recognition process.

2 Model based Object Class Detection

The key challenge of **Object Detection** is the ability to recognize any member in a category of objects in spite of wide variations in visual appearance due to geometrical transformations, change in viewpoint, or illumination. To deal with these challenges, we developed a novel 3D feature model based object class detection method. Our objective is to detect the object given an arbitrary 2D view using a general 3D feature model of the class. Here the objects can be arbitrarily transformed (with translation and rotation), and the viewing position and orientation of the camera is arbitrary as well. In addition, camera parameters are assumed to be unknown.

Object detection in such a setting has been considered a very challenging problem due to various difficulties of geometrically modeling relevant 3D object shapes and the effects of perspective projection. In our work, we exploit a recently proposed 3D reconstruction method using homographic framework for 3D object shape reconstruction. Given a set of 2D images of an object taken from different viewpoints around the object with unknown camera parameters, which are called model views, the 3D shape of this specific object can be reconstructed using the homographic framework proposed in [10]. In our method, 3D shape is represented by a volume consisting of binary slices with 1 denoting the object and 0 for background. By using this method, we can not only reconstruct 3D shapes for the objects to be detected, but also have access to the homographies between the

2D views and the 3D models, which are then used to build the 3D feature model for object class detection.

In the feature modeling phase of our method, SIFT features [12] are computed for each of the 2D model views and mapped to the surface of the 3D model. Since it is difficult to accurately relate 2D coordinates to a 3D model by projecting the 3D model to a 2D view (with unknown camera parameters), we propose to use a homography transformation based algorithm. Since the homographies have been obtained during the 3D shape reconstruction process, the projection of a 3D model can be easily computed by integrating the transformations of slices from the model to a particular view, as opposed to directly projecting the entire model by estimation of the projection matrix. To generalize the model for object class detection, images of other objects of the class are used as supplemental views. Features from these views are mapped to the 3D model in the same way as for those model views. A codebook is constructed from all of these features and then a 3D feature model is built. The 3D feature model thus combines the 3D shape information and appearance features for robust object class detection.

Given a new 2D test image, correspondences between the 3D feature model and this testing view are identified by matching feature. Based on the 3D locations of the corresponding features, several hypotheses of viewing planes can be made. For each hypothesis, the feature points are projected to the viewing plane and aligned with the features in the 2D testing view. A confidence is assigned to each hypothesis and the one with the highest confidence is then used to produce the object detection result.

2.1 Research Background and Related Works

As the approaches for recognizing an object class from some particular viewpoint or detecting a specific object from an arbitrary view are advancing toward maturity [3, 9, 11], solutions to the problem of object class detection using multiple views are still relatively far behind. Object detection can be considered even more difficult than classification, since it is expected to provide accurate location and size of the object.

Researchers in computer vision have studied the problem of multi-view object class detection resulting successful approaches following two major directions. One path attempts to use increasing number of local features by applying multiple feature detectors simultaneously [1, 6, 13–15]. It has been shown that the recognition performance can be benefited by providing more feature support. However, the spatial connections of the features in each view and/or between different views have not been pursued in these works. These connections can be crucial in object class detection tasks. Recently, much attention has been drawn to the second direction related to multiple views for object class detection [5, 7, 8]. The early methods apply several single view detectors independently and combine their responses via some arbitration logic. Features are shared among the different single-view detectors to limit the computational overload. Most recently, Thomas *et al*. [16] developed a single integrated multi-view detector that accumulates evidence from different training views. Their work combines a multi-view specific object recognition system [9], and the Implicit Shape Model for object class detection [11], where single-view codebooks are strongly connected by the exchange of information via sophisticated activation links between each other.

Here we introduce a unified method to relate multiple 2D views based on 3D object modeling. The main advantage of this method is an novel efficient object detection system capable of recognizing and localizing objects from the same class under different viewing conditions. Consequently, 3D locations of the features are considered during detection and better accuracy is obtained.

2.2 Object Detection using 3D Shape Model

3D Shape Model. Let I_i denote the foreground likelihood map (where each pixel value is the likelihood of that pixel being a foreground) in the ith view of total M views. Considering a reference plane, π_r, in the scene with homography $H_{\pi_r,i}$ from the ith view to π_r, warping I_i to π_r gives the warped foreground likelihood map:

$$\hat{I}_{i,r} = [H_{\pi_r,i}]I_i. \tag{1}$$

The visual hull intersection on π_r (AND-fusion of the shadow regions) is achieved by multiplying these warped foreground likelihood maps:

$$\theta_r = \prod_{i=1}^{M} \hat{I}_{i,r}, \tag{2}$$

where θ_r is the grid of the object occupancy likelihoods plane π_r. Each value in θ_r gives the likelihood of this grid location being inside the body of the object, indeed, representing a slice of the object cut out by plane π_r. It should be noted that due to the multiplication step in (2), the locations outside the visual hull intersection region will be penalized, thus, having a much lower occupancy likelihood.

The grid of the object occupancy likelihood can be computed at an arbitrary number of planes in the scene with different heights, each giving a slice of the object. Naturally this does not apply to planes that do not pass through the object's body, since visual hull intersection on these planes will be empty, therefore a separate check is not necessary.

Let $\mathbf{v}_x$, $\mathbf{v}_y$, and $\mathbf{v}_z$ denote the vanishing points for the X, Y, and Z directions, respectively, and $\mathbf{l}$ be the normalized vanishing line of reference plane in the XYZ coordinate space. The reference plane to the image view homography can be represented as

$$\hat{H}_{ref} = \begin{bmatrix} \mathbf{v}_x & \mathbf{v}_y & \mathbf{l} \end{bmatrix}. \tag{3}$$

Supposing that another plane π has a translation of z along the reference direction Z from the reference plane, it is easy to show that the homography of plane π to the image view can be computed by

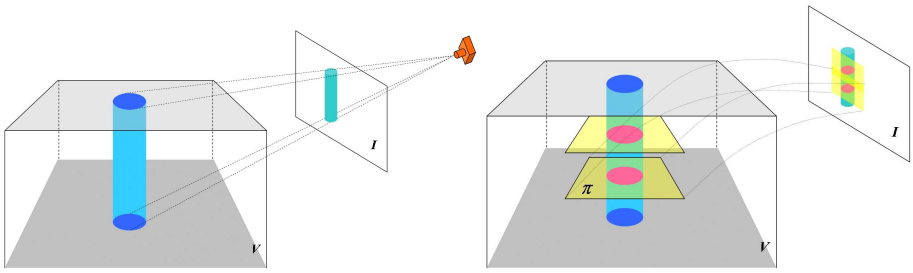
$$\hat{H}_{\pi} = \begin{bmatrix} \mathbf{v}_x & \mathbf{v}_y & \alpha z \mathbf{v}_z + \mathbf{l} \end{bmatrix} = \hat{H}_{ref} + [\mathbf{I}|\alpha z \mathbf{v}_z], \tag{4}$$

where α is a scaling factor. The image to plane homography H_{π} is obtained by inverting $\hat{H}_{\pi}$.

Starting with a reference plane in the scene (typically the ground plane), visual hull intersection is performed on successively parallel planes in the up direction along the body of the object. The occupancy grids θ_i are stacked up to create a three dimensional data structure $\Theta = [\theta_1; \theta_2; \ldots \theta_M]$. Θ represents a discrete sampling of a continuous occupancy space encapsulating the object shape. Object structure is then segmented out from Θ by dividing the space into the object and background regions using the geodesic active contour method [2]. By using the above homography based framework, 3D models for different objects can be constructed. In our method, not only the 3D shape of the target object is exploited, but also the appearance features. We relate the features with the 3D model to construct a *feature model* for object class detection.

The features used in our work are computed using the SIFT feature detector [12]. Feature vectors are computed for all of the training images. In order to efficiently relate the features computed from different views and different objects, all the detected features are attached to the 3D surface of the previously built model. By using the 3D feature model, we avoid storing all the 2D training views, thus there is no need to build complicated connections between the views. The spatial relationship between the feature points from different views are readily available, which can be easily retrieved when matched feature points are found.

The features computed in 2D images are attached to the 3D model by using the novel homographic framework. Instead of directly finding the 3D location of each 2D feature, we map the 3D



(a) Volume projection (b) Plane transformation

Fig. 1. Illustration of equivalence of 3D to 2D projection and plane transformation using homographies. (a) A 2D view of a 3D volume V is generated by projecting the volume on a image plane. (b) The same view can be obtained by integrating the transformation of each slice in the volume to the image plane using homographies.

points from the model's surface to the 2D views, and find the corresponding features. Our method does not require the estimation of a projection matrix from 3D model to a 2D image plane, which is a non-trivial problem. In our work, the problem is successfully solved by transforming the model to various image planes using homography. Since the homographies between the model and the image planes have already been obtained during the construction of the 3D model, we are able to map the 3D points to 2D planes using homography transformation.

In our work, a 3D shape is represented by a binary volume V, which consists of K slices S_j, $j \in [1, K]$. As shown in Fig. 1(b), each slice of the object is transformed to a 2D image plane by using the corresponding homography $\hat{H}$ in (4). The transformed slice accounts for a small patch of the object projection. Integrating all these K patches together, the whole projection of 3D object in the 2D image plane can be produced. In this way, we obtain the model projection by using a series of simple homography transformations and the hard problem of estimating the projection matrix of a 3D model to a 2D view is avoided.

In our method, the 3D shapes are represented using binary volumes with a stack of slices along the reference direction. Thus, the surface points can be easily obtained by applying edge detection techniques. After transforming the surface points to 2D planes, feature vectors computed in 2D can be related to the 3D points according to their locations. That is the way a 3D feature model is built.

The training images in our work come from two sources. One set of images is taken around a specific object of the target class to reconstruct it in 3D as shown in Fig. 2. These images are called *model views*, which provide multiple views of the object but are limited to the specific object. To generalize the model for recognizing other objects in the same class, another set of training images is obtained by using Google image search. Images of objects in the same class with different appearances and postures are selected. These images are denoted as the *supplemental views*.

Since the homographies between the supplemental images and the 3D model are unknown, features computed from the supplemental images cannot be directly attached to the feature model. Instead, we utilize the model views as bridges to connect the supplemental images to the model as illustrated in Fig. 2. For each supplemental image, the model view, which has the most similar viewpoint is specified. The supplemental images are deformed to their specified view by using an affine transformation alignment. Then we can assume that each supplemental image will have the same homography as the model's corresponding view. The 2D features computed from all of the supplemental training images can now be correctly attached to the 3D model surface using the same

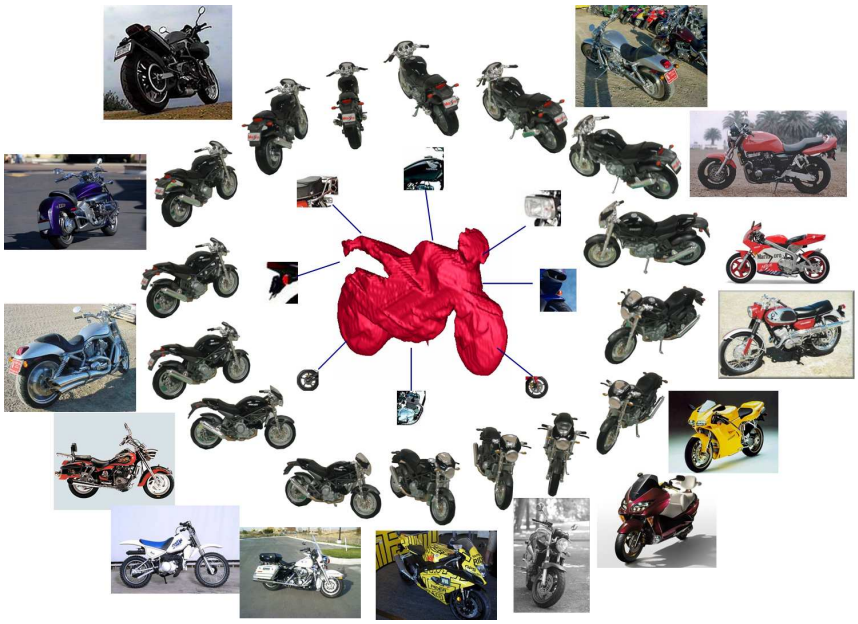


Fig. 2. Construction of 3D feature model for motorbikes. 3D shape model of motorbike (at center) is constructed using the model views (images on the inner circle) taken around the object from different viewpoints. Supplemental images (outer circle) of different motorbikes are obtained by using Google's image search. The supplemental images are aligned with the model views for feature mapping. Feature vectors are computed from all the training images and then attached to the 3D model surface by using the homography transformation.

method as discussed for the model views. A codebook is constructed by combining all the mapped features with their 3D locations.

Object Class Detection. Given a new test image, our objective is to detect objects belonging to the same class in this image by using the learnt 3D feature model M. Each entry of M consists of a code and its 3D locations $\{c, l_c^3\}$. Let F denote the SIFT features computed from the input image, which is composed by the feature descriptor and its 2D location in the image $\{f, l_f^2\}$. Object O_n is detected by matching the features F to the 3D feature model M.

In our work, feature matching is achieved in three phases. In the first phase, we match the features by comparing all the input features to the codebook entries in Euclidean space. However, not all the matched codebook entries in 3D are visible at the same time from a particular viewpoint. So, in the second phase, matched codes in 3D are projected to viewing planes and hypotheses of viewpoints are made by selecting viewing planes with the largest number of visible points projected. In the third phase, for each hypothesis, the projected points are compared to 2D matched feature points using both feature descriptors and locations. This is done by iteratively estimating the affine transformation between the feature point sets and removing the outliers with large distance between corresponding points. Outliers belonging to the background can be rejected during this matching

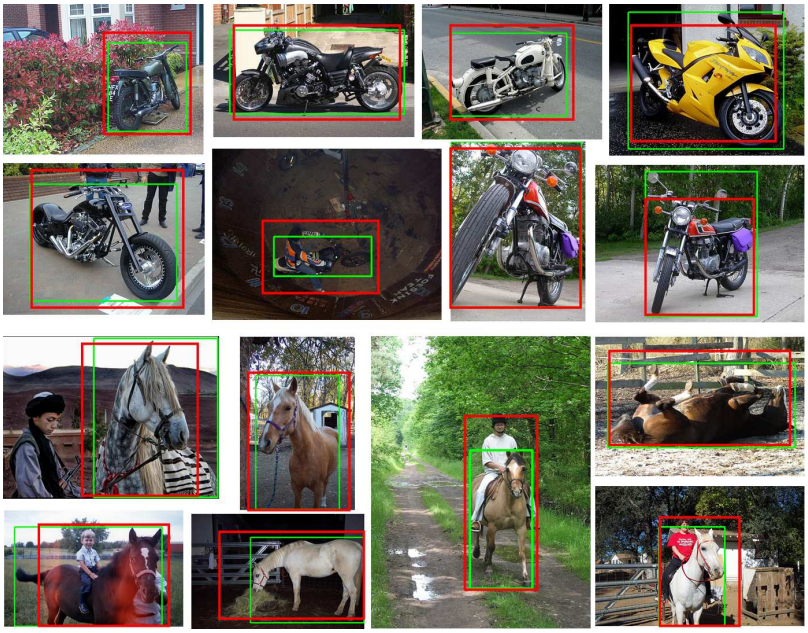


Fig. 3. Detection of motorbikes and horses using the proposed approach. The ground truth is shown in green and red boxes display our detected results.

process. The object location and bounding box is then determined according to the 2D locations of the final matched feature points. The confidence of detection is given by the degree of match.

Experimental Results. Our method has been tested on two object classes: motorbikes and horses. For the motorbikes, we took 23 model views around a motorbike and obtained 45 supplemental views by using Google's image search. Some training images of the motorbikes and the 3D shape model are shown in Fig. 2. For the horses, 18 model views were taken and 51 supplemental views were obtained.

To measure the performance of our 3D feature model based object class detection technique, we have evaluated the method on the PASCAL VOC Challenge 2006 test dataset [4], which has become a standard testing dataset for objective evaluation of object classification and detection algorithms. The dataset is very challenging due to the large variability in the scale and poses, the extensive clutter, and poor imaging conditions. Some successful detection results are shown in Fig. 3. The green box indicates the ground truth, while our results are shown in red boxes.

For quantitative evaluation, we adopt the same evaluation criteria used in PASCAL VOC challenge, so that our results can be directly comparable with [4, 8, 16]. By using this criteria, a detection is considered correct, if the area of overlap between the predicted bounding box B_p and ground truth bounding box B_{gt} exceeds 50% using the formula

$$\frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} > 0.5. \tag{5}$$

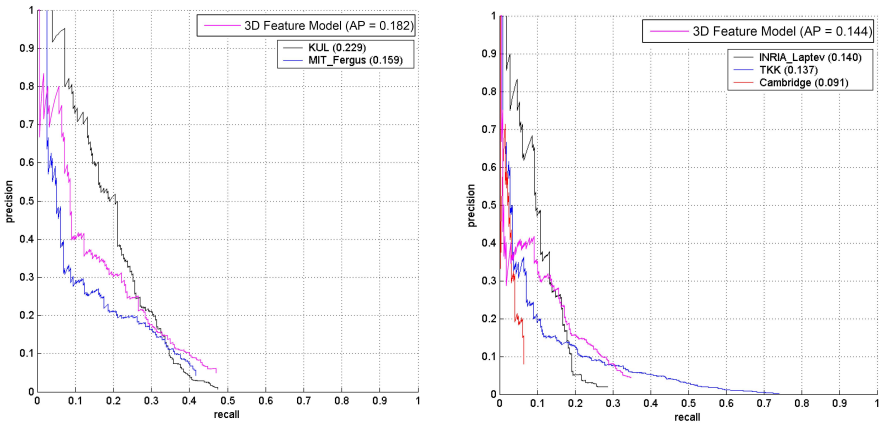


Fig. 4. The PR curves for (a) motorbike detection and (b) horse detection using our 3D feature model based approach. The curves reported in [4] on the same test dataset are also included for comparison.

The *average precision* (AP) and *precision-recall* (PR) curve can then be computed for performance evaluation.

Fig. 8(a) shows the PR curves of our approach and the methods in [8,16] for motorbike detection. The curve of our approach shows a substantial improvement over the precision compared to the method in [8], which is also indicated by the AP value (0.182). Although our performance is lower than that of [16], considering the smaller training image set used in our experiments, this can be regarded as satisfactory. Fig. 8(b) shows the performance curves for horse detection. While there is no result reported in the VOC challenge using researchers' own training dataset for this task, we compared our result to those using the provided training dataset. Our approach performs better than the reported methods and obtained AP value of 0.144. It is noted that the absolute performance level is lower than that of motorbike detection, which might be caused by the non-rigid body deformation of horses.

3 Correlation Pattern Recognition Based on View Morphing

In this report, we describe two new methods for synthesizing new views of a known object. We have shown previously that given a set of paired signatures in **Y** and **X**, it is possible to model the view synthesis process as a linear transform *A* such that **AY** = **X**. In fact, the minimum squared error solution was shown to be

$$\mathbf{A} = \mathbf{X}\mathbf{Y}^T \left(\mathbf{Y}\mathbf{Y}^T\right)^{-1}. \tag{6}$$

Then, if y (a column of Y) is an "observed" signature, the matrix **A** can be used to obtain the prediction **x** (a column of **X**) via the equation **Ay** = **x**.

3.1 Interpolated Nearest Neighbor

The problem is that while the overall squared error is minimized, the performance at or near individual signatures can be poor. To overcome this, we limit the signature pairs to be the nearest-neighbors

of the input signature. Thus, given an observed input signature (say $\mathbf{z}$), we select the $\mathbf{M}$ nearest neighbor columns of $\mathbf{Y}$ (denoted by the matrix $\mathbf{Z}$) and the corresponding columns of $\mathbf{X}$ (now denoted by the matrix $\mathbf{U}$) and write the equation:

$$\mathbf{AZ} = \mathbf{U}. \tag{7}$$

The problem now is that $\mathbf{M}$ is substantially smaller than the dimension of the vector space, and the minimum squared error solution for $\mathbf{A}$ cannot be computed since $\mathbf{ZZ^T}$ is singular. One approach to overcome this limitation is to avoid the explicit computation of $\mathbf{A}$ altogether by assuming that the observed input signature $\mathbf{y}$ can be approximated as a weighted linear combination (or interpolation) of its nearest neighbors, i.e.

$$\mathbf{y} = \mathbf{Za}. \tag{8}$$

Of course, given $\mathbf{y}$ and $\mathbf{Z}$ it is easy to obtain the weights

$$\mathbf{a} = \left(\mathbf{Z}^T\mathbf{Z}\right)^{-1}\mathbf{Zy}. \tag{9}$$

With this simplification, the estimation equation becomes

$$\mathbf{Ay} = \mathbf{AZa} = \mathbf{Ua}. \tag{10}$$

In other words, if the interpolation weight vector $\mathbf{a}$ is known, the predicted response to $\mathbf{y}$ is simply the interpolation (weighted linear combination) of the columns of $\mathbf{U}$. In other words, we don't actually need to know the transform matrix $\mathbf{A}$, as long as we have an estimate for the weight vector a and the ideal predicted signatures in $\mathbf{U}$.

3.2 Constrained Optimization

In this section we combine the advantages of the original LMSE approach, and the nearest neighbor interpolation technique. Specifically, we require $\mathbf{A}$ to minimize MSE across all the data, but satisfy exact relations in the neighborhood of the test vector.

Let Y be the input data matrix and X be the desired output data. We wish to find the linear transform $\mathbf{A}$ that will satisfy the equation in a minimum squared error (mse) sense. In terms of the columns of the input and output data matrices, this can be written as

$$E = \sum_{i=1}^{N} \left|\mathbf{A}\mathbf{y}_i - \mathbf{x}_i\right|^2. \tag{11}$$

At the same time, we wish to exactly satisfy the linear relation in the immediate neighborhood of the test input $\mathbf{z}$. Let the columns of $\mathbf{Z}$ represent the $\mathbf{M}$ nearest neighbors of $\mathbf{z}$, and the corresponding exact outputs are the columns of $\mathbf{U}$. There we have the hard constraints $\mathbf{AZ} = \mathbf{U}$. We find $\mathbf{A}$ by solving for one row at a time by formulating a constrained minimization problem. Let $\mathbf{a^T}$ be a row of $\mathbf{A}$, and be the corresponding row of $\mathbf{X}$. The error can be written for each row as

$$\begin{aligned} E &= \left|\mathbf{a}^T\mathbf{Y} - \mathbf{x}^T\right|^2 \\ &= \left|\mathbf{Y}^T\mathbf{a} - \mathbf{x}\right|^2 = \mathbf{a}^T\mathbf{Y}^T\mathbf{Ya} + \mathbf{x}^T\mathbf{x} - 2\mathbf{a}^T\mathbf{Yx} \\ &= \mathbf{a}^T\mathbf{Da} + \mathbf{x}^T\mathbf{x} - 2\mathbf{a}^T\mathbf{Yx}. \end{aligned} \tag{12}$$

The constraints on $\mathbf{a^T}$ is similarly written as

$$\mathbf{Z}^T\mathbf{a} = \mathbf{u} \tag{13}$$

where $\mathbf{u}$ is the corresponding row of $\mathbf{U}$.

Using the method of Lagrange multipliers, we form the functional

$$\Phi = \mathbf{a}^T\mathbf{D}\mathbf{a} + \mathbf{x}^T\mathbf{x} - 2\mathbf{a}^T\mathbf{Y}\mathbf{x} - 2\lambda_1\left(\mathbf{a}^T\mathbf{z}_1 - u_1\right) - 2\lambda_2\left(\mathbf{a}^T\mathbf{z}_2 - u_2\right) - \cdots - 2\lambda_M\left(\mathbf{a}^T\mathbf{z}_M - u_M\right). \quad (14)$$

The derivative of this w.r.t. to **a** is

$$\nabla_{\mathbf{a}}\Phi = 2\mathbf{D}\mathbf{a} - 2\mathbf{Y}\mathbf{x} - 2\left(\lambda_1\mathbf{z}_1 + \lambda_2\mathbf{z}_2 + \cdots + \lambda_1\mathbf{z}_1\right) = 2\mathbf{D}\mathbf{a} - 2\mathbf{Y}\mathbf{x} - 2\mathbf{Z}\mathbf{l}. \quad (15)$$

Setting this to zero, we get

$$\mathbf{D}\mathbf{a} = \mathbf{Y}\mathbf{x} + \mathbf{Z}\mathbf{l} \quad (16)$$

or

$$\mathbf{a} = \mathbf{D}^{-1}\left(\mathbf{Y}\mathbf{x} + \mathbf{Z}\mathbf{l}\right). \quad (17)$$

We then substitute this in the constraint equation to get

$$\mathbf{u} = \mathbf{Z}^T\mathbf{D}^{-1}\left(\mathbf{Y}\mathbf{x} + \mathbf{Z}\mathbf{l}\right) = \mathbf{Z}^T\mathbf{D}^{-1}\mathbf{Y}\mathbf{x} + \mathbf{Z}^T\mathbf{D}^{-1}\mathbf{Z}\mathbf{l}. \quad (18)$$

Therefore,

$$\mathbf{l} = \left(\mathbf{Z}^T\mathbf{D}^{-1}\mathbf{Z}\right)^{-1}\left(\mathbf{u} - \mathbf{Z}^T\mathbf{D}^{-1}\mathbf{Y}\mathbf{x}\right). \quad (19)$$

Finally, the expression for a is obtained as

$$\mathbf{a} = \mathbf{D}^{-1}\mathbf{Y}\mathbf{x} + \mathbf{D}^{-1}\mathbf{Z}\left(\mathbf{Z}^T\mathbf{D}^{-1}\mathbf{Z}\right)^{-1}\left(\mathbf{u} - \mathbf{Z}^T\mathbf{D}^{-1}\mathbf{Y}\mathbf{x}\right). \quad (20)$$

3.3 Examples and Comparisons

In this section we illustrate and compare the previous LMSE technique, the interpolated nearest neighbor method, and the new constrained optimization approach. We present several cases below where a group of four images on the left show the performance of the prediction algorithms, and the input test image is shown on the right. In each case, the the top left corner is the "ideal" image to be predicted. The top right is the result of the previous LMSE approach, the bottom left is the output of the nearest neighbor method, and the bottom right is the prediction obtained using the constrained technique. For each predicted image, the normalized similarity to the ideal image is shown in the title (larger is better). For example in case 1 below, the LMSE approach achieves a similarity of 0.75, and the nearest neighbor approach produces a worse result with similarity of 0.55. The constrained optimization technique performs the best by producing an image which has 0.79 similarity to the ideal image. The same is found to be true for the other 3 cases as well where the constrained optimization technique performs the best.

References

1. A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *CVPR (1)*, pages 26–33, 2005.
2. V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.
3. H. Chang and D.-Y. Yeung. Graph laplacian kernels for object classification from a single example. In *CVPR (2)*, pages 2011–2016, 2006.

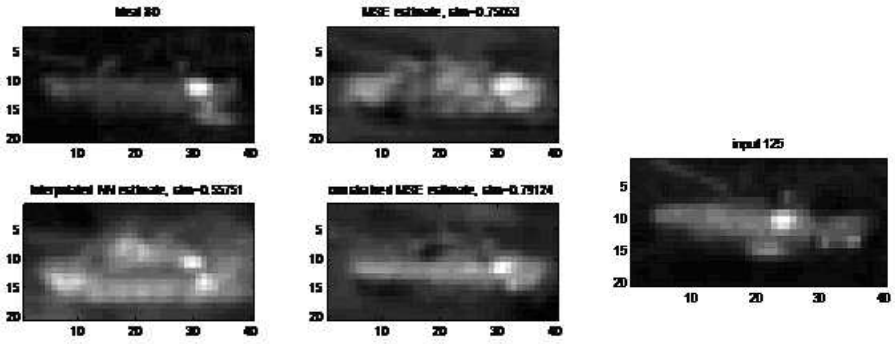


Fig. 5. Case 1.

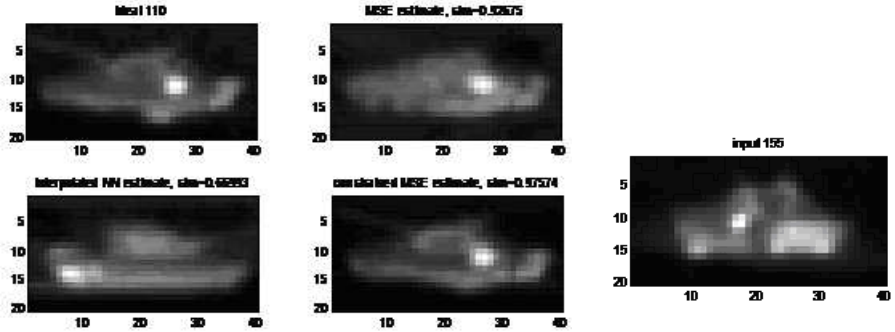


Fig. 6. Case 2.

4. M. Everingham, A. Zisserman, C. Williams, and L. van Gool. The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results. http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf.
5. J. D. R. Farquhar, D. R. Hardoon, H. Meng, J. Shawe-Taylor, and S. Szedmák. Two view learning: SVM-2K, theory and practice. In *NIPS*, 2005.
6. L. Fei-Fei, R. Fergus, and P. Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *ICCV*, pages 1134–1141, 2003.
7. R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *ICCV*, pages 1816–1823, 2005.
8. R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *CVPR*, pages 443–461, 2005.
9. V. Ferrari, T. Tuytelaars, and L. Van Gool. Integrating multiple model views for object recognition. In *CVPR*, volume 2, pages 105–112, 2004.
10. S. M. Khan, P. Yan, and M. Shah. A homographic framework for the fusion of multi-view silhouettes. In *ICCV*, 2007.
11. B. Leibe and B. Schiele. Scale-invariant object categorization using a scale-adaptive mean-shift search. In *DAGM*, pages 145–153, 2004.
12. D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, Nov. 2004.

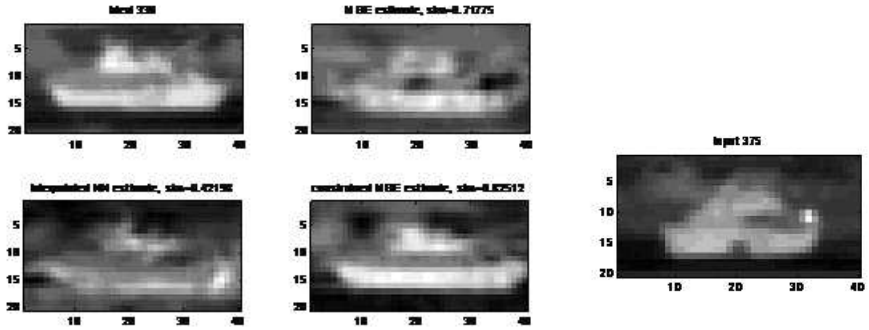


Fig. 7. Case 3.

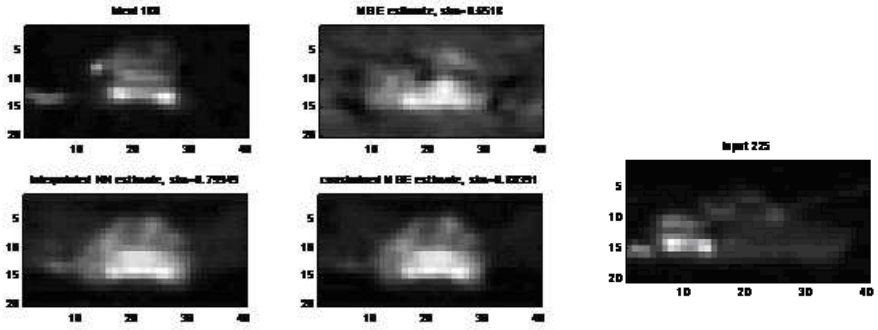


Fig. 8. Case 4.

13. E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *ECCV (4)*, pages 490–503, 2006.
14. J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *ICCV*, pages 503–510, 2005.
15. E. B. Sudderth, A. B. Torralba, W. T. Freeman, and A. S. Willsky. Learning hierarchical models of scenes, objects, and parts. In *ICCV*, pages 1331–1338, 2005.
16. A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. Van Gool. Towards multi-view object class detection. In *CVPR*, volume 2, pages 1589–1596, 2006.